

“Software Engineering”
Course
a.a. 2018-2019
Template version 1.0
Deliverable #3

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

**Dashboard Monitoraggio
Ambientale**

Date	20/01/2019
Deliverable	3
Team (Name)	Orange Justice

Team Members		
Name & Surname	Matriculation Number	E-mail address
Francesco Perilli	250424	francesco.perilli96@gmail.com
Giulio Marcantonio	249201	giulio.marcantonio@gmail.com
Loris Savini	247293	Saviniloris01@gmail.com
Raffaele Limongi	247977	raffaele9700@gmail.com
Alice Di Bernardo	249176	alicedibernardo@gmail.com

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

REQUISITI GENERICI	6
<i>Detailed Scenarios</i>	<i>6</i>
<i>Requisiti Funzionali</i>	<i>7</i>
Diagrammi Use Case.....	10
Descrizione estesa degli Use Case	11
Requisiti della GUI	23
Requisiti della business logic	24
Requisiti del database	24
<i>Requisiti Non Funzionali</i>	<i>25</i>
<i>Requisiti esclusi</i>	<i>25</i>
<i>Assunzioni</i>	<i>26</i>
<i>Priorità requisiti</i>	<i>27</i>
 ARCHITETTURA SOFTWARE	 27
<i>Component Diagram</i>	<i>28</i>
<i>Sequence Diagram</i>	<i>30</i>
 ER DESIGN	 32
 CLASS DIAGRAM OF THE IMPLEMENTED SYSTEM	 34
 DESIGN DECISION.....	 36
 FRs AND NFRs ARE SATISFIED BY DESIGN	 38
 EFFORT RECORDING	 42

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
Manutenzione del database	20/11/2018	22/11/2018	Abbiamo deciso che il DB, una volta al mese, dovrà necessariamente effettuare una pulizia dei vecchi dati per far spazio ai nuovi senza compromettere l'experience del gestore. L'orario scelto sarà di notte e verrà quindi effettuata una manutenzione automatica ogni mese.
Gestione del massimale	27/11/2018	29/11/2018	Il massimale potrà essere gestito e salvato dal gestore che è responsabile dei sensori/del sensore della sua zona. Il gestore quindi avrà a disposizione un'area in cui potrà gestire il massimale del sensore impostando un nuovo valore.
MYSQL o NoSQL?	10/12/2018	12/12/2018	Abbiamo discusso molto su quale tecnologia utilizzare, ed è venuto fuori che all'inizio adotteremo MYSQL e non NOSQL come deciso all'inizio. Ci siamo accorti che a livello di tempistiche costa di più NOSQL, quindi prima inizieremo con MYSQL e, nel caso in cui fosse poco efficiente, andremo ad implementare il NOSQL.

Gestione sensori non funzionanti	12/12/2018	15/12/2018	<p>Come gestiamo i sensori non funzionanti? Per alleggerire il carico dei dati, abbiamo deciso che se un sensore non è funzionante è inutile che acquisisca dati (errati, non veritieri) per poi inviarli.</p> <p>Così facendo andremmo ad occupare solo spazio inutile per dati nulli o non veritieri. Quindi se un sensore non è funzionante verrà inserito in una lista (lato software) in attesa di essere riparato o sostituito.</p>
Problema nella colorazione di una singola cella nel caso in cui il valore ha superato il massimale	22/12/2018	22/12/2018	<p>Abbiamo risolto utilizzando il metodo “setcellfactory”, che ci ha permesso di controllare se il valore acquisito supera il massimale impostato e successivamente di colorare soltanto la cella relativa al valore di quel sensore.</p>
Problema nella simulazione per l’invio frequente dei dati	06/01/2019	09/01/2019	<p>Abbiamo risolto utilizzando le timeline proposte da javafx, che ci permettono di avviare una porzione di codice che permette l’invio frequente dei dati, quando il sensore ha superato più di 3 volte il massimale stabilito.</p>

A. Requirements Collection

A.0 Detailed Scenarios

- **Upload di un valore acquisito dal sensore:**
 1. Il sensore acquisisce un nuovo valore (ogni minuto).
 2. Il valore acquisito viene gestito dalla parte logica del sistema per il check di eventuali errori.
 3. Dopo il check, la parte logica invia il dato rilevato per la memorizzazione sul database.
 4. Il database viene aggiornato aggiungendo una nuova riga nella tabella dato.
 5. La dashboard ricarica i nuovi dati presenti sul database.

- **Monitoraggio sensori**
 1. Il gestore inserisce il codice nella form di login.
 2. Clicca sul pulsante “login” per effettuare l’accesso.
 3. In base al tipo di gestore viene caricata una precisa homepage:
 - 3.1 Se il gestore è di tipo **area**, viene caricata una home dove sarà possibile selezionare prima l’area, poi il comune e successivamente l’edificio da monitorare.
 - 3.2 Se il gestore è di tipo **zona**, viene caricata una home dove sarà possibile selezionare il comune in cui può monitorare i dati, successivamente l’edificio a cui è interessato.
 - 3.3 Se il gestore è di tipo **edificio**, viene caricata una home dove sarà possibile selezionare solo l’edificio di cui è responsabile.
 4. Il gestore clicca sull’edificio che vuole monitorare.
 5. Dopo la scelta dell’edificio, la dashboard si apre e il gestore visualizza in una tabella tutti i sensori presenti in quell’edificio con i relativi dati.

- **Cambio del massimale di un sensore**
 1. Il gestore inserisce il codice nella form di login.
 2. Clicca sul pulsante “login” per effettuare l’accesso.
 3. In base al tipo di gestore viene caricata una precisa homepage:
 - 3.1 Se il gestore è di tipo **area**, viene caricata una home dove sarà possibile selezionare prima l’area, poi il comune e successivamente l’edificio da monitorare.
 - 3.2 Se il gestore è di tipo **zona**, viene caricata una home dove sarà possibile selezionare il comune in cui può monitorare i dati, successivamente l’edificio a cui è interessato.
 - 3.3 Se il gestore è di tipo **edificio**, viene caricata una home dove sarà possibile selezionare solo l’edificio di cui è responsabile.
 4. Il gestore clicca sull’edificio che vuole monitorare.

5. Dopo la scelta dell'edificio, la dashboard si apre e il gestore visualizza in una tabella tutti i sensori presenti in quell'edificio con i relativi dati.
6. Clicca sul pulsante “modifica” presente nella colonna “mod_massimale” del sensore scelto.
7. Viene aperta una nuova scheda per la modifica del massimale.
8. Il gestore clicca sulla textfield presente nella scheda e inserisce il nuovo valore.
9. Preme sul tasto salva per applicare le modifiche.
10. Il valore viene passato alla parte logica del sistema che effettua l'update nel database.
11. Il gestore viene re-indirizzato alla dashboard.

A.1 Functional Requirements

- **Login da parte del gestore alla dashboard**

Il gestore della dashboard, avrà la possibilità di effettuare il login alla dashboard per il monitoraggio ambientale attraverso un codice univoco in relazione al tipo di gestore che sia.

Quindi ad ogni gestore, appartenente ad una diversa zona, verrà assegnato un codice univoco che gli permetterà di accedere ai dati, in base alla zona in cui ha bisogno di prelevare tali informazioni, che vengono inviati dai sensori installati.

Verrà considerato anche che il codice univoco sarà diverso per le varie città e che quindi il gestore della città A (ad esempio) non potrà accedere alle informazioni inviate dai sensori di un'altra città B (ad esempio).

- **Selezione dell'area da parte di un gestore**

Il gestore, dopo aver effettuato il login, avrà la possibilità di selezionare l'area in cui vuole accedere per la visualizzazione dei dati che verranno inviati dai sensori installati.

Ogni gestore potrà selezionare un comune/edificio attraverso una tableview, ma con una diversa presentazione dei dati in base al tipo di gestore che è, ad esempio il gestore dell'area potrà selezionare un qualsiasi edificio di qualsiasi comune, in cui sono presenti dei sensori, presente nell'area stabilita, mentre ad esempio un gestore di un edificio avrà la possibilità di selezionare solo l'edificio, in quel determinato comune, in cui è responsabile.

Il gestore dell'area quindi, potrà selezionare qualsiasi zona, per poter controllare e visionare i dati dei sensori.

Il gestore del comune invece, potrà selezionare qualsiasi edificio presente in quel determinato comune per poter controllare e visionare i dati.

Infine il gestore dell'edificio avrà la possibilità di selezionare solo l'edificio a cui appartiene e a cui può accedere ai dati dei sensori.

- **Visualizzazione dei dati inviati dai sensori**

Un gestore loggato al sistema, avrà a disposizione una sezione in cui potrà visualizzare i dati periodicamente inviati dal sensore.

Potrà accedere a tale sezione cliccando su uno degli edifici proposti in base al tipo di gestore che è.

I dati inviati dal sensore periodicamente, verranno quindi mostrati in quest'area predisposta per il gestore.

Il sistema effettuerà un refresh dei dati ogni minuto.

- **Visualizzazione dello stato dei sensori**

Un gestore loggato al sistema, attraverso una sezione potrà visualizzare lo stato del funzionamento di un sensore/dei sensori presente nell'edificio, nel comune o nell'area in cui si trovano, quindi un gestore dell'edificio A del comune A non potrà visualizzare lo stato del sensore dell'edificio B del comune A ad esempio, soltanto il gestore del comune potrà farlo.

Il gestore, potrà facilmente capire lo stato di funzionamento del sensore/dei sensori attraverso un quadrato (colorato), affianco al nome del sensore, predisposto nella sezione per la visualizzazione dello stato del sensore/sensori.

- **Gestione del massimale**

Un gestore loggato al sistema, dopo aver effettuato il login con successo, potrà gestire nella dashboard, il massimale/i massimali relativi ai sensori di cui è responsabile e di cui vuole visionare i dati. Una volta scelto il massimale per il sensore/sensori, tale valore/valori verranno settati nel sensore così da poter confrontare subito i valori del massimale stabilito con quelli che acquisisce in tempo reale.

- **Visualizzazione degli alert**

Il gestore loggato al sistema, potrà accedere ad una casella per la visualizzazione dei messaggi di alert, mandati dal sistema e relativi al funzionamento del sensore e alla notifica del superamento del massimale da più sensori di più variabili ambientali.

Tale casella verrà quindi posizionata all'interno della dashboard, dove si potrà accedere ai messaggi di alert attraverso un click su un button che aprirà una sezione dedicata a tali messaggi di alert.

E' previsto l'inserimento di un altro button per poter cancellare i messaggi una volta letti e quindi per pulire la casella dei messaggi di alert.

- **Acquisizione dei dati ambientali**

Il sensore o i vari sensori installati negli edifici, all'aperto e nelle stanze, appartenenti ad una determinata zona/comune, acquisiranno i valori ambientali presenti in tale area.

Questi valori verranno acquisiti minuto per minuto e inviati opportunamente per permettere la visualizzazione dei dati al gestore che accede a tale dashboard di quella determinata area.

- **Invio dei dati ambientali**

Il sensore/i sensori dopo aver acquisito i valori ambientali delle stanze/edifici, effettueranno un invio di tali valori ad una database per la memorizzazione e la successiva visualizzazione nella dashboard, ma allo stesso tempo verrà fatto un backup in locale per permettere comunque la visualizzazione dei dati al gestore responsabile di quell'area, ma cosa più importante per evitare che, in caso di malfunzionamento del database, i dati raccolti dai sensori vengano persi.

- **Invio dello stato di funzionamento**

Il sensore/i sensori, oltre all'invio dei dati ambientali acquisiti, invieranno il proprio stato di funzionamento per permettere al gestore di verificare, tramite la dashboard, se il sensore/sensori stanno effettivamente funzionando in quel momento e quindi stanno acquisendo dei valori reali.

Ovviamente lo stato del funzionamento del sensore/sensori verranno salvati all'interno del database.

- **Controllo del massimale**

Il sensore, dopo aver acquisito le variabili ambientali, dovrà effettuare un check sul massimale, infatti se i valori ambientali acquisiti sono superiori al massimale stabilito, il sensore dovrà inviare più frequentemente i dati (ogni 30 secondi), invece se il massimale non viene superato il sensore invia i dati una volta al minuto.

- **Gestione back-end del sistema**

L'admin, dopo aver effettuato il login, tramite una UI specifica, potranno aggiungere/rimuovere aree, edifici, luoghi aperti, comuni, sensori, gestori ecc..

L'admin quindi sarà in grado di gestire in back-end tutto il sistema apportando le relative modifiche per aggiungere/rimuovere gli oggetti sopra descritti.

A1.1 Use Case Diagrams

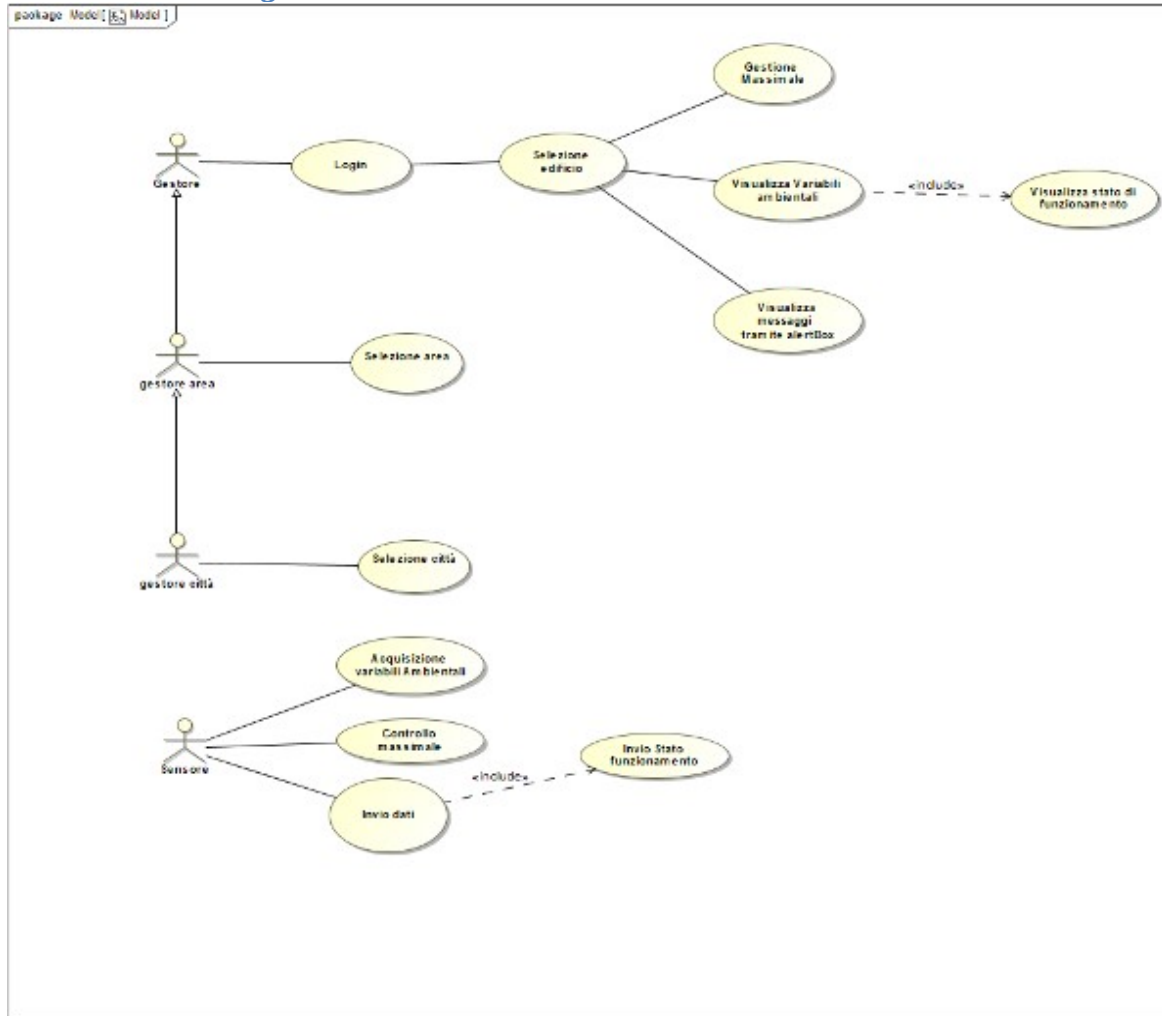


Figura1

A1.2 Tabular description of the most relevant use case

USE CASE 1	LOG IN	
Goal in Context	Accesso per le aree riservate alla gestione dei sensori	
Scope & Level	Primary Task	
Preconditions	Presenza del codice personale per effettuare il Log In	
Success End Condition	Si ha accesso alla dashboard per il monitoraggio dei dati	
Failed End Condition	Accesso alla dashboard negato!	
Primary, Secondary Actors	Gestore generico, dashboard	
Trigger	Inserimento del codice e login button	
DESCRIPTION	Step	Action
	1	Immettere il codice di accesso nell'apposito form
	2	Premere il Log In Button per mandare il codice al database per effettuare il controllo
	3	Accesso alla dashboard in caso di successo
EXTENSIONS	Step	Branching Action
	2a	Il codice inserito è errato e non ha trovato riscontro nel db 2a1 Compare un messaggio di errore
SUB-VARIATIONS		Branching Action
	3	La dashboard propone contenuti diversi in base al tipo del gestore che accede

RELATED INFORMATION	1 Log In
Priority:	Alta
Performance	Molto Rapido
Frequency	Una volta per ogni sessione
Channels to actors	Interattivo,database
Due Date	1.0

USE CASE 2	Selezione edificio	
Goal in Context	Il gestore clicca sull'edificio, presente in una lista, che vuole monitorare	
Scope & Level	Primary task	
Preconditions	Aver effettuato il Log In con successo	
Success End Condition	Si visualizzano i dati dei sensori presenti nell'edificio selezionato	
Failed End Condition	Non si visualizzano I dati dei sensori presenti nell'edificio selezionato	
Primary, Secondary Actors	Gestore, gestore area, gestore città dashboard	
Trigger	Click sull'edificio presente nella lista mostrata dopo l'accesso	
DESCRIPTION	Step	Action
	1	Aspetto il caricamento della dashboard
	2	La dashboard carica la lista dell'edificio/edifici in cui si possono visualizzare i dati
	3	Click sull'edificio che si desidera monitorare

EXTENSIONS	Step	Branching Action
	2a	Non vengono mostrati edifici disponibili al monitoraggio
SUB-VARIATIONS		Branching Action
	2	La dashboard carica 2a1 la selezione dell'area (usecase 3) 2a2 la selezione della città (usecase 4)

RELATED INFORMATION	2 Selezione edificio
Priority:	Alta
Performance	Il tempo per la richiesta al db e il caricamento dei dati nella lista contenente tutti i sensori dell'area selezionata
Frequency	Sempre
Channels to actors	Database
OPEN ISSUES	Caricamento lento?
Due Date	1.0

USE CASE 3	Selezione area
Goal in Context	Il gestore clicca sull'edificio, presente in una lista, che vuole monitorare
Scope & Level	Primary task
Preconditions	Aver effettuato il Log In con successo
Success End Condition	Si accede alla lista degli edifici
Failed End Condition	Non si riesce ad accedere alla lista degli edifici

Primary, Secondary Actors	Gestore area, gestore città dashboard	
Trigger	Click sull'area presente nella lista mostrata dopo l'accesso	
DESCRIPTION	Step	Action
	1	Aspetto il caricamento della dashboard
	2	La dashboard carica la lista dell'area in cui si possono visualizzare i dati
	3	Click sull'area che si desidera monitorare
EXTENSIONS	Step	Branching Action
	2a	Non vengono mostrate aree disponibili al monitoraggio
SUB- VARIATIONS		Branching Action
	2	La dashboard carica
		2a1 la selezione della città(usecase 4)

RELATED INFORMATION	3 Selezione area
Priority:	Alta
Performance	Il tempo per la richiesta al db e il caricamento dei dati nella lista contenente tutti gli edifici dell'area selezionata
Frequency	Sempre
Channels to actors	database
Due Date	1.0

USE CASE 4	Selezione città
Goal in Context	Il gestore della città clicca sulla città, presente in una lista,

	che vuole monitorare	
Scope & Level	Primary task	
Preconditions	Aver effettuato il Log In con successo	
Success End Condition	Si accede alla lista delle aree presenti	
Failed End Condition	Non si riesce ad accedere alla lista delle aree	
Primary, Secondary Actors	Gestore città dashboard	
Trigger	Click sulla città presente nella lista mostrata dopo l'accesso	
DESCRIPTION	Step	Action
	1	Aspetto il caricamento della dashboard
	2	La dashboard carica la lista delle città che si possono monitorare
	3	Click sulla città che si desidera monitorare
EXTENSIONS	Step	Branching Action
	2a	Non vengono mostrate città disponibili al monitoraggio

RELATED INFORMATION	4 Selezione città
Priority:	Alta
Performance	Il tempo per la richiesta al db e il caricamento dei dati nella lista contenente tutte le aree della città selezionata
Frequency	Sempre
Channels to actors	database
Due Date	1.0

USE CASE 5	Gestione del massimale	
Goal in Context	Il gestore modifica il valore del massimale relativo al sensore scelto cliccando sul button “save” per salvare la sua scelta	
Scope & Level	Secondary task	
Preconditions	Login,Selezione area	
Success End Condition	La modifica del massimale avviene con successo e il valore acquisisce un nuovo massimale	
Failed End Condition	La modifica del massimale fallisce e quindi il valore non varia	
Primary, Secondary Actors	Gestore,dashboard,sensore	
Trigger	Inserire il massimale e scegliere il sensore a cui applicare tale scelta. Alla fine della compilazione cliccare su salva.	
DESCRIPTION	Step	Action
	1	Click sul pulsante in alto a dx per accedere alla gestione del massimale
	2	Compilare il campo del valore del massimale e scegliere il sensore a cui applicarlo
	3	Cliccare sul button salva
	4	Ritorno automatico alla home e invio del nuovo valore al sensore
EXTENSIONS	Step	Branching Action
	3a	Il processo di salvataggio non va a buon fine 3a1 viene mostrato un messaggio d’errore
SUB-		Branching Action

VARIATIONS		
	4	Ritorno automatico alla homepage ma l'invio del nuovo valore fallisce

RELATED INFORMATION	5 Gestione del massimale
Priority:	Media
Performance	Molto veloce, bassa
Frequency	Raramente
Channels to actors	interactive
OPEN ISSUES	Il gestore può modificare più volte il valore di seguito il massimale?
Due Date	2.0

USE CASE 6	Visualizza variabili ambientali
Goal in Context	Il gestore potrà controllare le variabili ambientali acquisite dal sensore e il suo stato di funzionamento
Scope & Level	Primary task
Preconditions	Aver effettuato il Log In con successo ed aver selezionato l'area
Success End Condition	Si accede ad una dashboard contenente dei valori da monitorare
Failed End Condition	Viene restituita una dashboard non contenente dei valori reali (o nulli)
Primary, Secondary Actors	Gestore, Dashboard, Sensore
Trigger	Click su un edificio presente dalla lista dopo la selezione dell'area

DESCRIPTION	Step	Action
	1	Click sull'edificio da monitorare
	2	Viene aperta una nuova scheda contenente la lista dei sensori di quell'edificio scelto da monitorare
EXTENSIONS	Step	Branching Action
	2a	Viene visualizzato lo stato di funzionamento
SUB-VARIATIONS		Branching Action
	2	Viene aperta la dashboard contenente una lista vuota o con i valori dei sensori nulli

RELATED INFORMATION	6 Visualizza variabili ambientali
Priority:	Alta
Performance	Tanto, il tempo di richiesta al db e visualizzazione dei dati
Frequency	Sempre
Channels to actors	database
OPEN ISSUES	null
Due Date	1.0

USE CASE 7	Visualizza messaggi tramite alertbox
Goal in Context	Il gestore potrà controllare i messaggi urgenti, emessi dal sensore, come stato di funzionamento o più sensori che superano il massimale, attraverso un alertbox.
Scope & Level	Secondary Task
Preconditions	Aver effettuato il Log In con successo ed aver selezionato l'area
Success End	Si accede ad una nuova scheda contenente tutti gli alert più

Condition	urgenti e importanti	
Failed End Condition	Viene restituita una scheda vuota, senza messaggi	
Primary, Secondary Actors	Gestore, sensore	
Trigger	Click sulla casella dell'alertbox per visualizzare i messaggi	
DESCRIPTION	Step	Action
	1	Click sulla casella, con scritto alertbox, in alto a destra della home
	2	Viene aperta una nuova scheda contenente i messaggi più importanti che il sensore ha emesso
EXTENSIONS	Step	Branching Action
	2a	Viene visualizzato il codice del sensore che ha emesso il messaggio/i
SUB-VARIATIONS		Branching Action
	2	Viene visualizzata una scheda vuota senza alert

RELATED INFORMATION	7 Visualizza messaggi tramite alertbox
Priority:	Media
Performance	Il tempo di caricamento degli alert nella lista
Frequency	Spesso
Channels to actors	database
OPEN ISSUES	Nel caso in cui il sensore risulti sempre essere fuori uso? come lo gestiamo l'alert?
Due Date	2.0

USE CASE 8	Acquisizione variabili ambientali	
Goal in Context	Il gestore potrà controllare i messaggi urgenti, emessi dal sensore, come stato di funzionamento o più sensori che superano il massimale, attraverso un alertbox.	
Scope & Level	Primary Task	
Preconditions	Sensore funzionante	
Success End Condition	Il sensore acquisisce un valore ambientale	
Failed End Condition	Il sensore non riesce ad acquisire un valore ambientale	
Primary, Secondary Actors	Sensore,	
Trigger	Il sensore autonomamente ogni minuto acquisisce un nuovo valore ambientale	
DESCRIPTION	Step	Action
	1	Il sensore parte ed acquisisce il valore ambientale in modo autonomo
	2	Preleva il dato e si prepara per l'invio al database
SUB-VARIATIONS		Branching Action
	1	Il sensore non riesce ad acquisire il dato

RELATED INFORMATION	8 Acquisizione variabili ambientali
Priority:	Alta
Performance	Veloce, tempo di acquisizione della variabile ambientale
Frequency	Una volta al minuto
OPEN ISSUES	null

Due Date	1.0
----------	-----

USE CASE 9	Invio dati	
Goal in Context	Invio dei dati al database	
Scope & Level	Primary task	
Preconditions	Sensore funzionante	
Success End Condition	Invio dei dati al database è avvenuto correttamente	
Failed End Condition	L'invio dei data al database non è avvenuto correttamente	
Primary, Secondary Actors	Sensore. Database	
Trigger	Il sensore invia il dato acquisito al database	
DESCRIPTION	Step	Action
	1	Il sensore avvia la connessione al db e attende una risposta
	2	Invia i dati al database in caso di connessione riuscita
EXTENSIONS	Step	Branching Action
	1a	Risposta negativa 1a1 Il sensore non invia i dati al db
	2a	Risposta positiva 2a1 Viene inviato anche lo stato di funzionamento
SUB-VARIATIONS		Branching Action
	1	Il sensore riceve una risposta negativa e riprova dopo 1 minuto

RELATED INFORMATION	9 Invio dati
Priority:	Alta
Performance	Veloce, connessione al db+invio dei dati
Frequency	Una volta al minuto
Channels to actors	database
OPEN ISSUES	I dati arriveranno sempre in modo corretto al database?
Due Date	1.0

USE CASE 10	Controllo massimale	
Goal in Context		
Scope & Level	Primary task	
Preconditions	Valori/valore acquisiti	
Success End Condition	Il valore acquisito ha superato il massimale stabilito e necessita di essere inviato frequentemente	
Failed End Condition	Il valore acquisito non ha superato il massimale stabilito e non necessita di essere inviato frequentemente	
Primary, Secondary Actors	Sensore	
Trigger	Il sensore fa un controllo interno in cui valuta che il valore acquisito non superi il massimale	
DESCRIPTION	Step	Action
	1	Il sensore attende l'acquisizione del valore ambientale
	2	Memorizza il valore acquisito e fa un check con il massimale

EXTENSIONS	Step	Branching Action
	2a	Check del massimale 2a1 il valore ambientale ha superato il massimale, il sensore invia frequentemente il dato 2a2 il valore ambientale non ha superato il massimale 2a3 invio stato di funzionamento
SUB-VARIATIONS		Branching Action
	2	Viene gestito un valore nullo, il sensore non fa il check del massimale. Infine invia lo stato del sensore non funzionante.

RELATED INFORMATION	10 controllo massimale
Priority:	Alta
Performance	Molto veloce, check delle variabili acquisite
Frequency	Una volta ogni 30 secondi/una volta ogni minuto
Channels to actors	Database
OPEN ISSUES	Se il controllo avviene sempre con valori nulli(sensore non funzionante)?
Due Date	1.0

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

- Il Sistema deve prevedere una GUI per permettere l'accesso al sistema, dalla quale poter visualizzare i dati acquisiti dai sensori, il loro stato di funzionamento e i relativi messaggi importanti.

- La GUI deve presentare una dashboard diversa per ogni tipo di gestore che accede al sistema e quindi dovrà essere composta da più o meno dati, proprio in base al tipo di gestore che effettuerà l'accesso.
Quindi per ogni gestore che accede verrà presentata la dashboard associata.
- Il sistema, deve garantire la corretta navigazione delle schede dell'applicazione attraverso appositi pulsanti.
I button, proposti dal sistema, devono garantire il corretto funzionamento della navigazione da parte del gestore che accede al sistema, muovendosi avanti e indietro nella dashboard.

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

- Il sistema nella sua parte logica, dovrà avere una componente che si occupi di archiviare i dati ricevuti dai sensori per poterli memorizzare.
- Il sistema nella sua parte logica, dovrà avere una componente che si occupi di eseguire query sul database per poter far visualizzare correttamente le informazioni inviate dai sensori.
- Il sistema nella sua parte logica, dovrà avere una componente che si occupi di controllare l'accesso da parte dei gestori effettuando un controllo e successivamente richiamare la giusta dashboard.

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

- Il sistema deve prevedere lo storage delle variabili ambientali acquisite dai sensori e dello stato di funzionamento relativo al sensore, per garantire uno storage sicuro dei dati.
- Il database dovrà essere collegato alla parte logica del sistema per poter effettuare lo storage dei dati ed essere interrogato per la visualizzazione dei dati acquisiti dai sensori.

A.2 Non Functional Requirements

- **USABILITA'**

Il Sistema deve poter garantire una dashboard di buona fattura a livello grafico, ma soprattutto deve essere strutturata in modo semplice e di facile comprensione per garantire una user experience funzionale ma allo stesso tempo semplice.

La GUI, deve essere il piu' user-friendly possibile e funzionale.

L'interfaccia grafica verrà strutturata diversamente per ogni tipo di operazione(login, visualizzazione dei dati ecc..).

- **PERFORMANCE**

Il sistema dovrà essere in grado di gestire l'accesso parallelo di almeno 50 gestori e dovrà essere in grado di gestire fino a 150.000 sensori distribuiti per la città.

La dashboard inoltre dovrà aggiornarsi, autonomamente , almeno 1 volta al minuto per garantire il refresh dei dati per il monitoraggio dei sensori.

Il sistema quindi dovrà garantire un certo livello di performance al verificarsi dei casi sopra descritti.

- **SECURITY**

Il sistema deve poter garantire un minimo di sicurezza relativa ai dati che vengono mostrati.

Quindi ad ogni gestore viene data una chiave per accedere al sistema, così da evitare accessi non autorizzati al sistema.

Solo il gestore, in possesso della chiave di accesso di quell'edificio/zona, può accedere al sistema così da garantire la sicurezza dei dati(che vengono visualizzati solo da persone autorizzate) acquisiti dai sensori.

Si deve garantire anche che il personale non autorizzato acceda al sistema per andare ad effettuare modifiche al sensore(ad esempio modifiche relative al massimale).

A.3 Excluded Requirements

- **No gestione del sensore per il gestore**

Non è prevista nessuna funzionalità per gestire il sensore sia fisicamente e sia a livello software da parte del gestore.

Quindi il gestore non potrà gestire fisicamente il sensore ma dovrà chiamare un tecnico aziendale per effettuare ad esempio la rimozione del sensore nel caso in cui smetta di funzionare, oppure la sostituzione di un sensore.

Cosa più importante non verrà implementata una funzionalità per gestire a livello software il sensore, il gestore quindi potrà settare solo il valore del massimale e non potrà procedere con altre operazioni di modifica, come ad esempio la modifica dell'acquisizione del dato e il successivo invio, oppure la modifica su dove inviare il dato(quindi a quale database).

Il gestore inoltre non potrà aggiungere altre funzionalità al sistema/sensore e non potrà modificare quelle presenti.

- **No criptazione dei dati**

Non verrà gestita nessuna funzionalità riguardante la criptazione dei dati.

Il sistema non attuerà quindi delle misure adatte per criptare i dati durante l'invio dopo che il sensore ha acquisito tali valori, quindi il sensore dopo aver acquisito i dati non effettuerà una funzione a livello di criptazione dei dati, ma semplicemente invierà tali dati al database.

Il sistema quindi non gestirà questo tipo di funzionalità perché tali dati, dopo varie discussioni, non rientrano nella categoria di dati sensibili e quindi il sistema non effettuerà tali misure per garantire una protezione più efficiente.

A.4 Assumptions

- I diversi codici di accesso, utilizzati dai diversi gestori ,sono contenuti nel database.
- Il gestore, in base al tipo, visualizza una dashboard diversa con più o meno informazioni.
- Il gestore può anche modificare il massimale impostato.
- Un gestore di un edificio può monitorare i dati relativi solo ad un edificio a cui è assegnato
- Un gestore di città può monitorare i dati relativi a tutti gli edifici presenti in quella città(compresi i luoghi aperti) e assumiamo che ci sia più di un gestore di città
- Assumiamo che i sensori sono:
 1. Temperatura
 2. Umidità
 3. Pressione atmosferica(nel caso siano posti esternamente ad un edificio)
- I sensori sono simulati e quindi non fisicamente presenti.

- I sensori non possono essere interrogati dal sistema, ma autonomamente ogni sensore una volta al minuto invia i dati.
- I sensori controllano se il valore acquisito ha superato il massimale.
- La gestione dei sensori guasti non verrà effettuata dal gestore.
- Ogni sensore è identificato da un proprio codice *univoco* che lo distingue, anche in base a quale dato acquisisce.
- I dati acquisiti dai sensori vengono opportunamente immagazzinati in un database.

A.5 Prioritization

1. Il sistema deve essere performante.
 2. Acquisizione e invio dei dati
 3. Costruzione della dashboard in base alla tipologia del gestore.
 4. Costruzione della scheda di login
 5. Selezione dell'area da parte di un gestore
 6. Visualizzazione dei messaggi tramite alertbox
 7. Gestione del massimale
 8. Usabilità
-

B. Software Architecture

Come architettura software abbiamo scelto il pattern MVC(Model View Controller), affiancato dal seguente pattern architetturale DAO(Data Access Object) che è un design pattern che fornisce un'interfaccia astratta ad alcuni tipi di database tra cui MYSQL, mappando le chiamate della business logic sul modello di persistenza dati.

Il pattern MVC è uno dei più utilizzati per la programmazione ad oggetti, in grado di separare in modo efficace la logica di presentazione dei dati dalla logica di business e in grado di adattarsi ad ogni problema.

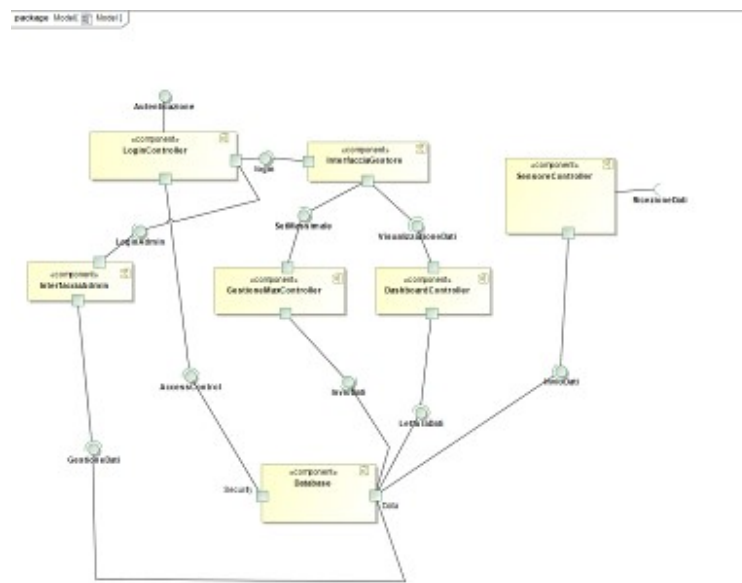
Il package Model, al suo interno, conterrà delle classi e degli oggetti che verranno manipolati dal controller per la visualizzazione dei dati.

Nella view, vengono gestiti gli insiemi degli elementi per la creazione dell'interfaccia grafica dell'applicazione utilizza dai gestori, utilizzeremo **JAVA FX** per realizzare tale grafica.

Il controller infine ha il compito di trasformare le interazioni dell'utente della view in azioni eseguite dal model.

Quindi al suo interno contiene le varie classi e metodi che permettono la gestione del sistema, le quali si occupano della manipolazione dei dati.

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



[Figura2](#)

Il Component Diagram è un diagramma che ha lo scopo di rappresentare la struttura interna del sistema software, modellando i componenti principali e le relazioni fra di essi. Nel dettaglio i componenti presenti nel sistema:

LoginController:

I dati che arrivano dall'interfaccia "Autenticazione", vengono processati da questa componente e successivamente inviati alla componente "**database**" per effettuare un controllo nella tabella del gestore.

InterfacciaGestore:

Sarà la parte del sistema accessibile dai gestori, essi dopo aver effettuato con successo la fase di login, potranno ricevere i dati acquisiti dai sensori e potranno

navigare all'interno della dashboard per monitorare oltre ai dati acquisiti, anche lo stato di funzionamento dei sensori.

DashboardController:

Questa componente permette di leggere i dati presenti nel database e caricarli sulla dashboard per permettere al gestore la visualizzazione di tali dati (come spiegato sopra).

Offre quindi la visualizzazione dei dati del sensore, dopo aver richiesto di leggere tali dati alla componente del database.

GestioneMaxController:

Questa componente invece, si occupa di settare il massimale inserito dal gestore.

Il nuovo valore aggiornato, viene passato alla componente Database per procedere allo storage.

Database:

Componente che si occupa di salvare sul server tutti i dati che il sistema gestisce, questo ci permette di mantenere i dati acquisiti dai sensori e di garantire un minimo di sicurezza per l'accesso al sistema.

SensoreController:

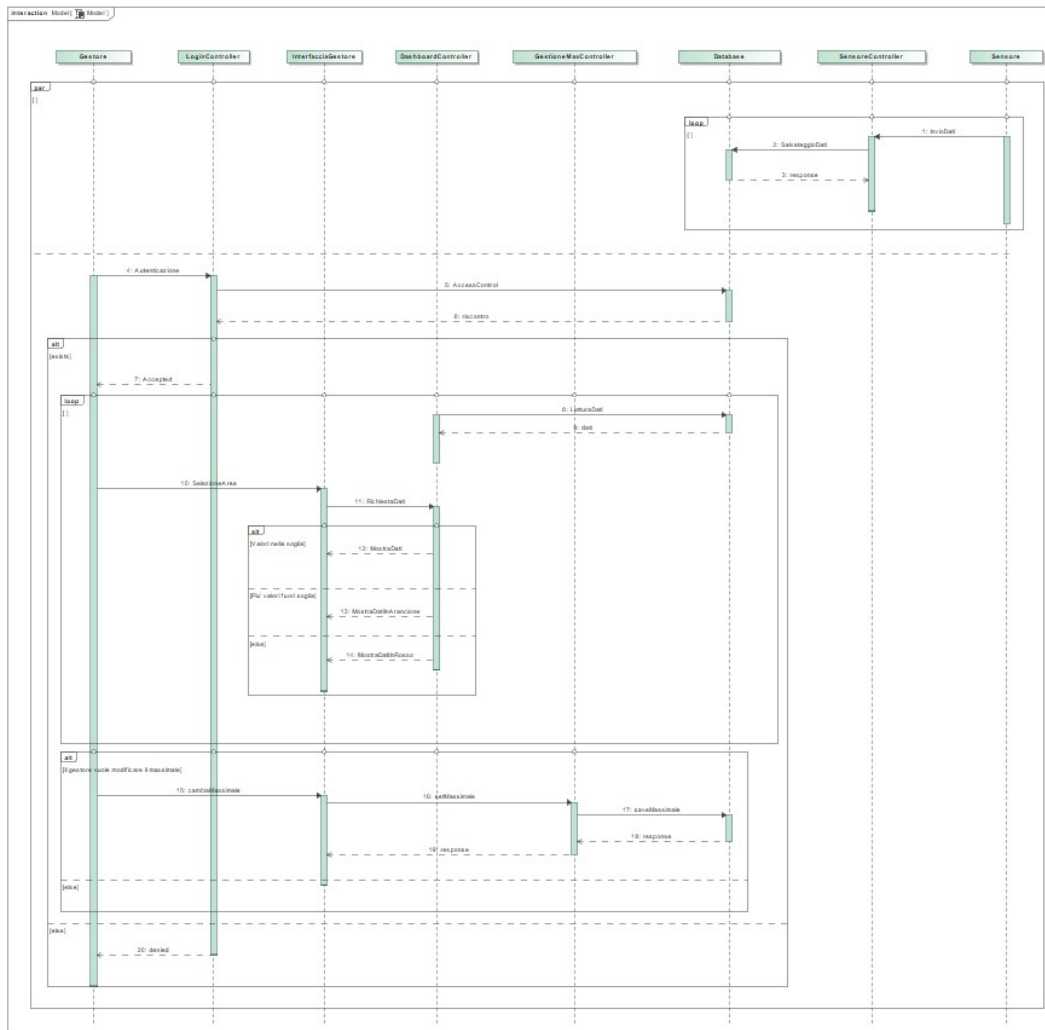
Questa componente si occupa di gestire i dati acquisiti dal sensore, elaborarli, controllarli ed infine inviarli alla componente database per la memorizzazione.

InterfacciaAdmin:

Come per "InterfacciaGestore", sarà la parte del sistema accessibile solo dagli admin, che potranno gestire in back-end tutto il sistema, aggiungendo e rimuovendo sensori, gestori e aree.

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

BE SURE THAT THE STRUCTURE IS SYNCHRONIZED WITH THE DYNAMIC VIEW

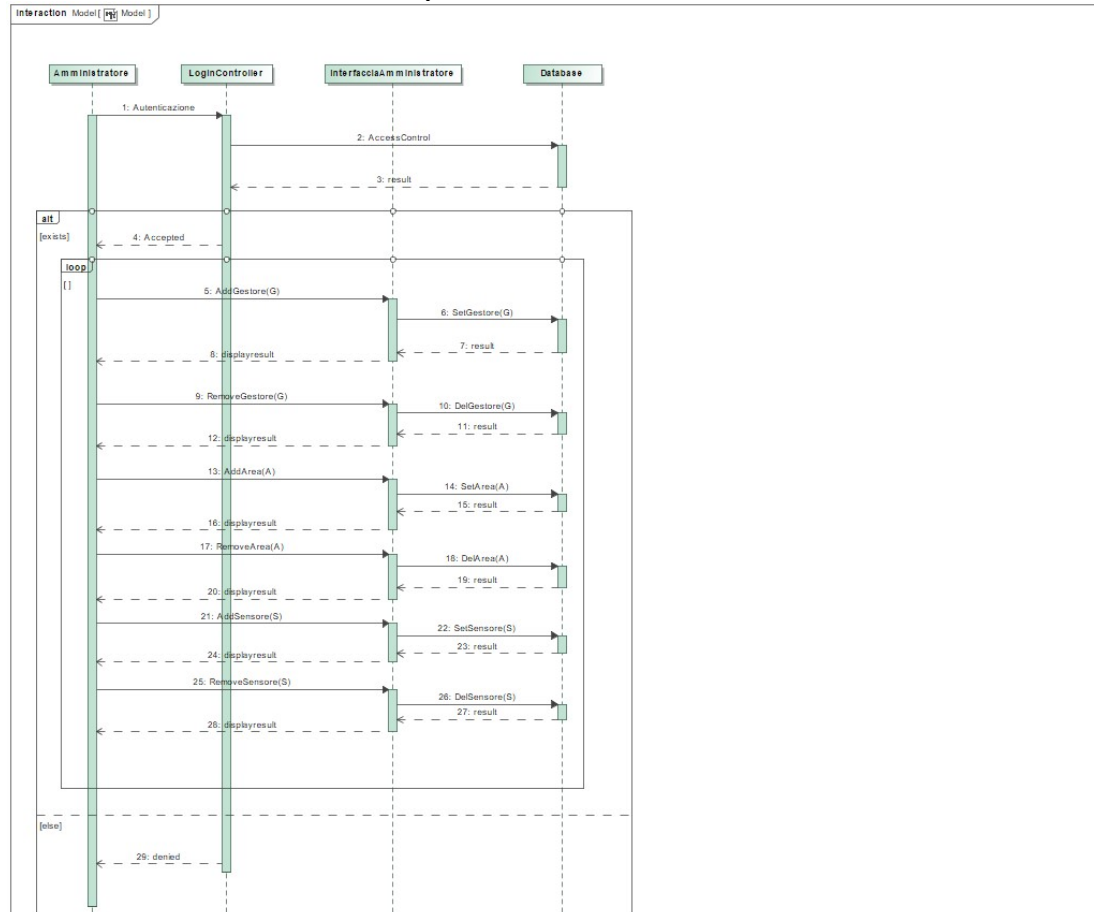


[Figura3](#)

Il Sequence Diagram è un diagramma utilizzato per descrivere uno scenario (una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate). I dati acquisiti dal sensore, vengono prelevati e gestiti dal SensoreController, che li invierà al database per effettuare quindi la memorizzazione di tali dati. Tutto ciò viene ripetuto in un loop infinito e quindi ogni minuto il sensore acquisisce nuovi dati, li passa al controller e quest'ultimo li invia al database per il salvataggio.

In parallelo il gestore accede al sistema autenticandosi, se risulta presente nel database potrà visualizzare i dati e gestire il massimale dei sensori, altrimenti non accede al sistema.

Nel caso in cui accede al sistema potrà visualizzare i dati dei sensori.

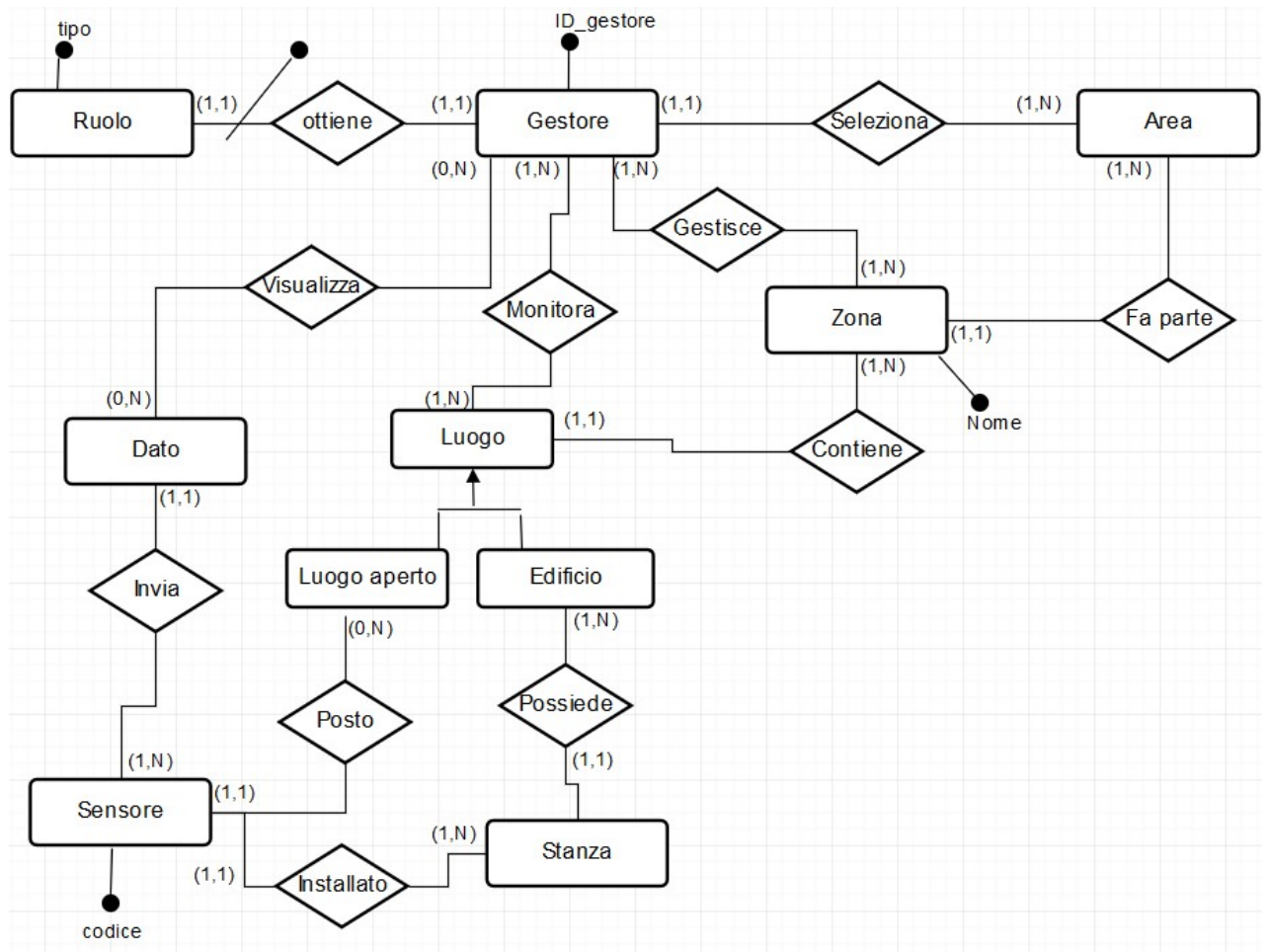


[Figura4](#)

L'admin accede al Sistema autenticandosi, se risulta presente nel database potrà gestire in back-end tutto il sistema.

Potrà quindi aggiungere/rimuovere dei sensori, dei gestori e delle aree selezionate, tramite appositi button e grazie ad una tableview che conterrà tutti i dati per effettuare tali operazioni.

C. ER Design



[*Figura5*](#)

Traduzione relazionale:

- Ruolo(**ID**, tipo, ID_gestore)
- Gestore(**ID**, codiceacc, nome, cognome, ID_area, ID_zona)
- Area(**ID**, nome, regione)
- Zona(**ID**, nome, ID_area)
- *Gestisce*(**ID_gestore**, **ID_zona**)
- Luogo(**ID**, nome, indirizzo, ID_zona, ID_gestore)
- *Monitora*(**ID_luogo**, **ID_gestore**)
- Luogo Aperto(**ID**, nome, indirizzo)
- Edificio(**ID**, nome, indirizzo)

- Stanza(**ID**, nome, piano, ID_edificio)
 - Sensore(**ID**, cod,stato,massimale, ID_luogoaperto, ID_edificio)
 - *Invia*(**ID sensore**, **ID dato**, datainvio)
 - Dato(**ID**, valore, tipo)
1. L'entità **Gestore** conserva i dati delle persone, coinvolte nel sistema:
gestore area, città, luogo(aperto/edificio)
 2. Un **gestore dell'area** seleziona l'area di competenza, può gestire 1 o piu' zone
che fanno parte di quell'area, può monitorare 1 o piu' edifici contenuti nella zona
di quell'area, può visualizzare 0 o piu' dati.
 3. Un **gestore di zona** può gestire 1 ed una sola zona, può monitorare 1 o piu'
edifici contenuti in quella zona, può visualizzare 0 o piu' dati.
 4. Un **gestore del luogo** può monitorare uno ed un solo luogo, può visualizzare 0 o
piu' dati.
 5. Un' **Area** può essere selezionata da piu' gestori, in un'area fanno parte 1 o piu
zone.
 6. Una **Zona** fa parte di un'unica area, può essere gestita da 1 o piu' gestori, una
zona può contenere 1 o piu' edifici.
 7. L'entità padre **Luogo** possiede due figli:
 - **Luogo aperto**, entità figlio Luogo che non possiede delle stanze(non ha match
con l'entità Stanze),
 - **Edificio**, entità figlio di Luogo che possiede delle stanze.Un **luogo** può essere monitorato da 1 o piu' gestori, un luogo è contenuto in una
sola zona.
In un **luogo aperto** possono essere posti 0 o piu' sensori.
Un **edificio** possiede almeno 1 o piu' stanze.
 8. Una **Stanza** è contenuta in un unico edificio, in una stanza possono essere
installati 1 o piu' sensori.
 9. Un **sensore** è posto in un unico luogo aperto, un sensore è installato in un'unica
stanza, Un sensore può inviare 1 o piu' dati.
 10. Un **Dato** è inviato da un'unico sensore, un dato può essere visualizzato da 0 o
piu' gestori.

D. Class Diagram of the implemented System

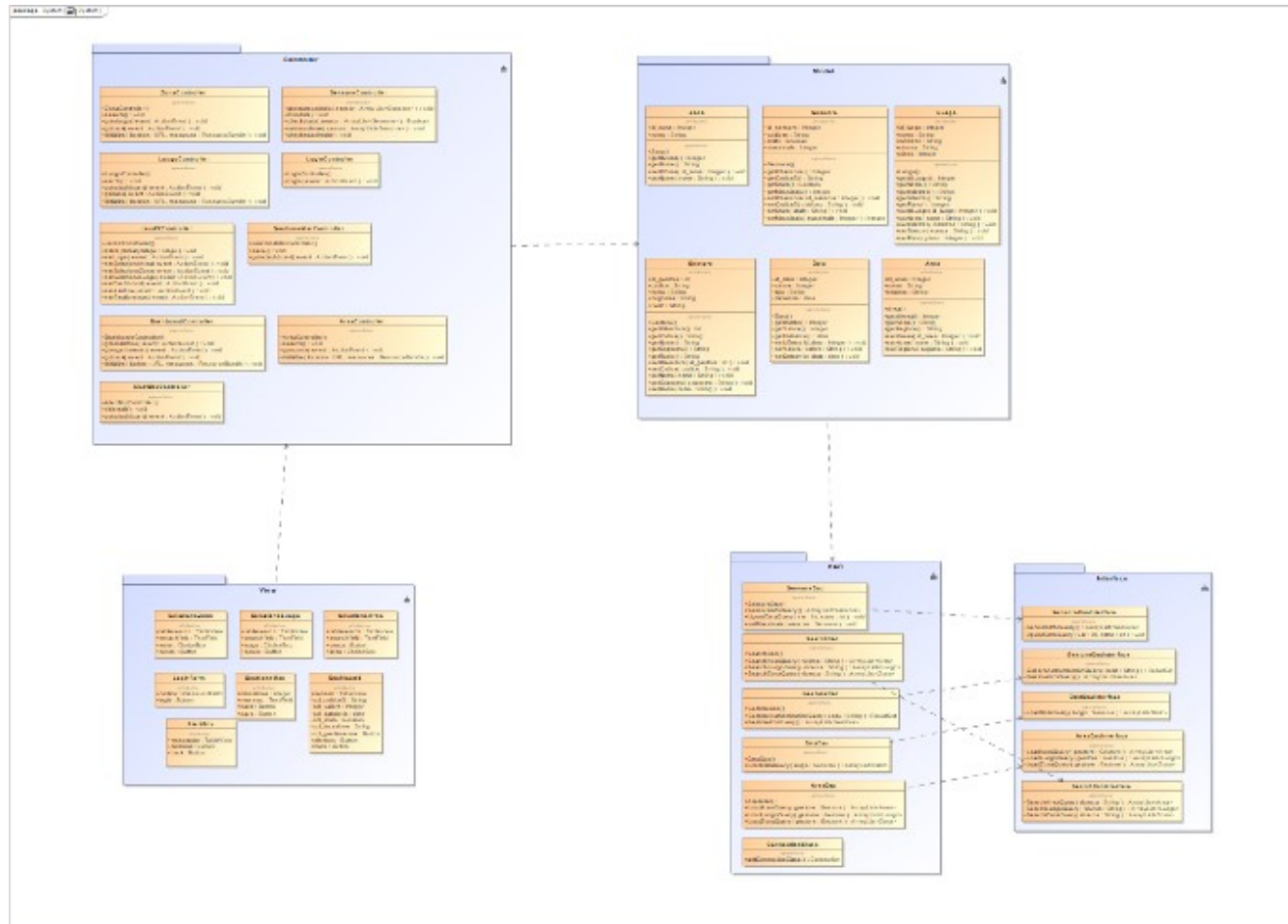


Figura6

Model:

La classe **Gestore** contiene, al suo interno, tutte le informazioni principali riguardanti il gestore e il suo ruolo nel sistema(area,zona,edificio).

La classe **Sensore**, contiene invece tutte le informazioni riguardanti il sensore: stato e codice del sensore(che identifica univocamente quel sensore).

La classe **Area**, contiene tutte le aree, caratterizzate da un nome e dalla regione di appartenenza, in cui sono installati i vari sensori.

Nella classe **VariabileAmbientale** abbiamo il dato acquisito dal sensore ogni minuto, il tipo di dato che è stato acquisito e la data e ora di invio di tale dato da parte del sensore.

La classe **Luogo** invece conserva al suo interno le informazioni riguardanti i vari edifici o luoghi aperti in cui sono installati i sensori, infatti viene conservato il nome del luogo, l'indirizzo in cui si trova, l'eventuale stanza(valido solo nel caso in cui sia un edificio chiuso), l'eventuale piano(valido solo nel caso in cui sia un edificio chiuso).

Infine la classe **Zona**, conserva le informazioni riguardanti le zone(comuni,frazioni,ecc..) che fanno parte di quella determinata area, contenendo quindi il nome di tale zone e la città di appartenenza.

View:

Nella **view** troviamo tutti gli oggetti relativi all'interfaccia grafica del sistema come:

- LoginForm,
- Dashboard,
- SelezioneArea,ecc...

Utilizzati quindi per far interfacciare il gestore al sistema.

Li utilizziamo per definire le varie azioni che poi i button,panel,form,passwordfield,ecc.. avranno quando il gestore interagirà con tali funzionalità.

Per esempio nella Login Form, mettiamo a disposizione un campo PasswordField per poter inserire il codice e un pulsante per poter effettuare l'accesso.

Controller:

Le classi del **controller** sono utilizzate per la gestione delle azioni svolte dal gestore come il login, la selezione dell'area, la visualizzazione dei dati, ecc..

Quindi in queste classi verranno definiti i metodi necessari per poter gestire le azioni effettuate dal gestore.

DAO:

Ogni classe definita nel dao contiene le query necessarie al controller per poter effettuare delle operazioni sul database.

Queste operazioni possono essere di vari tipi:

Ad esempio il controllo che avviene nella fase di login, dove il codice inserito dal gestore passa per il controller, che attraverso le query fornite dal dao, andrà a fare un check con i dati presenti nel database, per poter far autenticare correttamente il gestore all'interno del sistema., oppure per l'inserimento di nuovi dati nel database, ovvero l'invio dei dati acquisiti dal sensore.

Quindi in definitiva nel dao saranno presenti i metodi che permetteranno al controller, tramite le query, di fare interrogazioni al database.

E. Design Decisions

- **Dashboard diversa per i tipi di gestore**

Una delle prime decisioni a livello di design del sistema è stata proprio la gestione di diverse dashboard in base al tipo di ruolo del gestore.

Infatti abbiamo deciso che per ogni ruolo si ha una diversa dashboard, contenete piu' o meno dati in base proprio al tipo del gestore.

Abbiamo deciso di implementare così questa decisione per poter proporre una dashboard che faccia visualizzare solo lo stretto necessario per il tipo di gestore che la sta utilizzando.

Ad esempio al gestore dell'edificio è inutile far visualizzare delle informazioni superflue riguardanti ad esempio sensori posti in altre zone della città, perché comunque non sarà in grado di accedere a tali informazioni e quindi visualizzarle, perciò abbiamo deciso di sviluppare tre tipi di dashboard diverse per ogni gestore così da proporre le informazioni rilevanti e necessarie per ogni tipo di gestore, avendo così una dashboard minimale con lo stretto necessario a livello informativo per il gestore che andrà a visualizzare i dati.

- **Accesso sicuro al sistema**

Abbiamo deciso che per accedere al sistema bisogna inserire un codice univoco per l'identificazione del gestore e l'autorizzazione per la visualizzazione dei dati contenuti nella dashboard e che i sensori inviano periodicamente.

Questa pagina di login sarà quindi composta da un campo di testo dove sarà possibile inserire il codice univoco che verrà assegnato ad ogni tipo di gestore.

Questo codice ovviamente sarà diverso per ogni edificio in cui si vogliono controllare i dati dei sensori, quindi ad esempio il gestore dell'edificio A avrà un codice diverso rispetto al gestore dell'edificio B.

Abbiamo quindi deciso di implementare questa feature per garantire la giusta visualizzazione dei dati in base al tipo di gestore che accede, e che quindi si ricollega al primo punto descritto sopra, ovvero il gestore dopo l'accesso potrà utilizzare la dashboard contenente i dati in base al tipo di gestore che è, quindi il sistema, dopo l'autenticazione del gestore, caricherà la relativa dashboard con i relativi dati.

- **Scelta per la gestione dei messaggi del sensore tramite un AlertBox**

Abbiamo deciso di implementare nel nostro sistema un alertbox, che conterrà i messaggi di primaria importanza inviati dal sensore.

Questo alertbox verrà posizionato nella dashboard in alto a destra così da contenere tutti gli alert da parte dei sensori.

All'interno dell'alertbox, quindi, verrà visualizzata una lista contenente il codice univoco del sensore, la posizione in cui si trova e infine il tipo di problema che è stato notificato (funzionamento o massimale superato).

Infatti per un gestore della città riteniamo che sia più comodo ed efficiente l'utilizzo di un alertbox per la notifica di malfunzionamenti e di altri problemi, così da far evitare al gestore di visualizzare tutti i sensori di quella città cercando di capire quelli che non funzionano o i valori ambientali che hanno superato il massimale.

- **Scelta del database e linguaggio di programmazione**

Il linguaggio di programmazione utilizzato sarà java, perché abbiamo deciso di utilizzare tale linguaggio che abbiamo studiato di più in questi anni, accompagnandolo con javafx per la creazione grafica del sistema.

A livello grafico, anche con java, sarà possibile rispettare l'usabilità e quindi creare un applicativo semplice ma allo stesso tempo molto funzionale alle varie esigenze dei diversi gestori.

Per lo storage dei dati abbiamo deciso di utilizzare MySQL, che si connette in modo facile e veloce con java.

Abbiamo discusso molto su quale tipo di tecnologia utilizzare, e alla fine abbiamo deciso di implementare mysql piuttosto che NOSQL (anche se più performante).

Nel caso in cui mysql risulti essere poco performante, verrà implementato NOSQL.

Abbiamo deciso così, anche perché andremo ad effettuare mensilmente un dump del database e quindi il db verrà pulito una volta al mese.

- **Visualizzazione dei valori fuori soglia**

Nella lista dei sensori, mostrata al gestore, verrà inserito un pallino di un determinato colore (per ogni sensore), in base alla gravità del valore acquisito dai sensori.

Un sensore sarà verde nel caso in cui acquisisce un dato che non supera il massimale o rosso nel caso in cui il valore acquisito superi il massimale-

- 1) Edificio**

Anche nella lista degli edifici verrà mostrato graficamente un pallino, in base alla quantità di sensori che superano il massimale in quell'edificio.

Se:

- Tutti i sensori, presenti nell'edificio, acquisiscono un valore che non supera il massimale, verrà mostrato un pallino verde.

- Più' sensori (max il 35-40%) ,presenti nelle stanze di un edificio, acquisiscono un valore che supera il massimale, verrà mostrato un pallino **arancione**.
- La maggior parte dei sensori (piu del 35-40%) ,presenti nelle varie stanze, acquisiscono un valore che supera il massimale, verrà mostrato un pallino **rosso**.

2) Città

Anche nella lista delle città, verrà mostrato graficamente un pallino in base alla quantità degli edifici in cui i sensori superano il massimale.

Se:

- Tutti i sensori, presenti negli edifici di tale città, acquisiscono un valore che non supera il massimale, verrà mostrato un pallino **verde**.
- Più' sensori (max il 35-40%), presenti negli edifici di tale città, acquisiscono un valore che supera il massimale, verrà mostrato un pallino **arancione**.
- La maggior parte dei sensori (piu' del 35-40%), presenti negli edifici di tale città, acquisiscono un valore che supera il massimale, verrà mostrato un pallino **rosso**.

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

- **Functional Requirements**

1. *Il Sistema deve prevedere una GUI per permettere l'accesso al sistema, dalla quale poter visualizzare i dati acquisiti dai sensori, il loro stato di funzionamento e i relativi messaggi importanti.*

La GUI è stata realizzata tramite JavaFX.

La GUI è stata realizzata in modo da poter essere accessibile tramite un pc, installando un applicazione scritta in java e disegnata tramite il software applicativo : JavaFX.

I valori acquisiti dai sensori sono immagazzinati in un database e sono inviati, tramite queries, al client che accede alla dashboard.

Il lato applicativo del client si occupa di collocare tali dati nelle giuste posizioni della lista presente nella dashboard.

Così quindi, viene garantita la visualizzazione dei dati, dello stato di funzionamento del sensore, ma anche dei messaggi più importanti relativi ai sensori.

Viene garantita inoltre anche la fase di login (accesso al sistema).

- 2. La GUI deve presentare una dashboard diversa per ogni tipo di gestore che accede al sistema e quindi dovrà essere composta da più o meno dati, proprio in base al tipo di gestore che effettuerà l'accesso.
Quindi per ogni gestore che accede verrà presentata la dashboard associata.***

In base al tipo di gestore che accede, il sistema propone una GUI diversa, infatti se il gestore è di tipo “area”, viene proposta una GUI dove prima è possibile selezionare l’area e successivamente il luogo e l’edificio.

Invece se il gestore è di tipo “edificio”, viene proposta una GUI dove è possibile selezionare solo l’edificio di cui è responsabile.

Tutto ciò avviene grazie al “ruolo”, ottenuto dopo l’accesso del gestore, che viene salvato e passato alla classe che carica la relativa dashboard in base al tipo di gestore.

- 3. Il sistema, deve garantire la corretta navigazione delle schede dell'applicazione attraverso appositi pulsanti.
I button, proposti dal sistema, devono garantire il corretto funzionamento della navigazione da parte del gestore che accede al sistema, muovendosi avanti e indietro nella dashboard.***

Il codice Java ci permette di navigare all’interno della dashboard.

- 4. Il sistema nella sua parte logica, dovrà avere una componente che si occupi di archiviare i dati ricevuti dai sensori per poterli memorizzare.***

Abbiamo realizzato una classe in java che permette di generare(simulare l'acquisizione delle variabili ambientali)i dati e successivamente inviarli. Una classe si occuperà quindi di generare i dati(compreso lo stato di funzionamento).

- 5. *Il sistema nella sua parte logica, dovrà avere una componente che si occupi di eseguire queris sul database per poter far visualizzare correttamente le informazioni inviate dai sensori.***

Abbiamo realizzato una classe in java che si occupa di gestire ed inviare i dati al database.

Quindi ogni minuto esegue le relative queris per inserire i dati nelle apposite tabelle del database.

- 6. *Il sistema nella sua parte logica, dovrà avere una componente che si occupi di controllare l'accesso da parte dei gestori effettuando un controllo e successivamente richiamare la giusta dashboard.***

Abbiamo realizzato una classe in java che si occupa di gestire e controllare i dati inseriti nel form della login.

Quindi ogni volta che un gestore inserisce il codice per accedere, nella classe viene richiamata una queries che controlla(contenuta nell'apposito package DAO)se il codice inserito è contenuto nel database, successivamente richiamando la giusta dashboard per il tipo di gestore che accede.

- 7. *Il sistema deve prevedere lo storage delle variabili ambientali acquisite dai sensori e dello stato di funzionamento relativo al sensore, per garantire uno storage sicuro dei dati. Il database dovrà essere collegato alla parte logica del sistema per poter effettuare lo storage dei dati ed essere interrogato per la visualizzazione dei dati acquisiti dai sensori.***

Tramite il design pattern DAO siamo stati in grado di realizzare nella parte logica del sistema una componente in grado di interagire con il database per lo storage dei dati.

Abbiamo quindi utilizzato java JDBC per il collegamento tra il database e la parte logica del sistema.

- **NON FUNCTIONAL REQUIREMENTS**

- ❖ **Usabilità**

Per quanto riguarda *l'usabilità*, il sistema, che viene strutturato con il pattern MVC, sarà composto dal package view contenente tutte le schede (a livello grafico) che permettono di garantire tale requisito.

Infatti ogni file .xml sarà strutturato in modo semplice e user friendly per il gestore, inserendo graficamente le componenti adatte (liste, barra di ricerca, pulsanti per ricercare edifici-luoghi, pulsanti per tornare alle schermate precedenti e altro), ottenendo così un applicativo semplice ed efficiente che soddisfi il requisito di usabilità.

- ❖ **Performance**

L'obiettivo è quello di riuscire a rispettare tale tempo indicato per permettere il corretto refresh dei dati sulla dashboard, tutto ciò grazie anche alla divisione dei compiti tra le classi grazie al pattern MVC utilizzato.

Quindi ogni volta che il sensore acquisisce un dato e lo invia, la dashboard deve aggiornare i dati.

L'obiettivo è quello di riuscire a gestire contemporaneamente almeno 50 gestori che accedono al sistema.

Quindi il database utilizzato, dovrà essere in grado di rispondere nel minor tempo possibile a tutte le richieste contemporanee da parte dei gestori, inoltre dovrà essere in grado di gestire fino a 150.000 segnali al minuto inoltrati dai sensori e riuscire quindi ad effettuare tali queries in tempo breve. Così da garantire e non intaccare le performance del sistema.

I valori acquisiti verranno inseriti e gestiti in vari array (più sensori ci sono più array verranno utilizzati per alleggerire il carico), per poi prelevare i valori da inserire nel database, ogni volta che si ripete quest'operazione l'array verrà pulito.

- ❖ **Security**

Per garantire un minimo di sicurezza, il sistema propone graficamente, tramite una classe contenuta nel package view, il form per inserire il codice ed effettuare quindi il login.

Mentre la parte logica, una classe contenuta nel package controller, avrà il compito di prelevare i dati inseriti e controllare che tali dati siano presenti nel database, restituendo un messaggio di successo nel caso siano presenti, altrimenti un messaggio negativo che non permette di accedere al sistema e di utilizzare le sue funzionalità.

Così facendo si andrà a garantire la sicurezza, per evitare che persone non autorizzate possano visualizzare i dati e modificare il massimale di ogni sensore.

G. Effort Recording

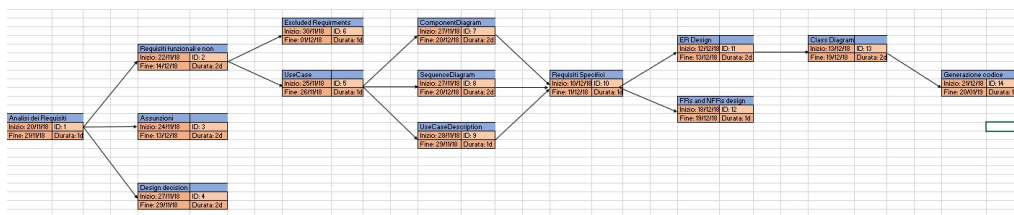


Figura7

OR

PertModel

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

Total Number spent: 131h

Doing: 118h

Learning: 13h

Appendix. Prototype

<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>

Di seguito qualche screenshots della dashboard:

Quella di seguito è la situazione che si presenterà al gestore quando aprirà l'app ad ogni avvio, ovvero la fase di autenticazione.

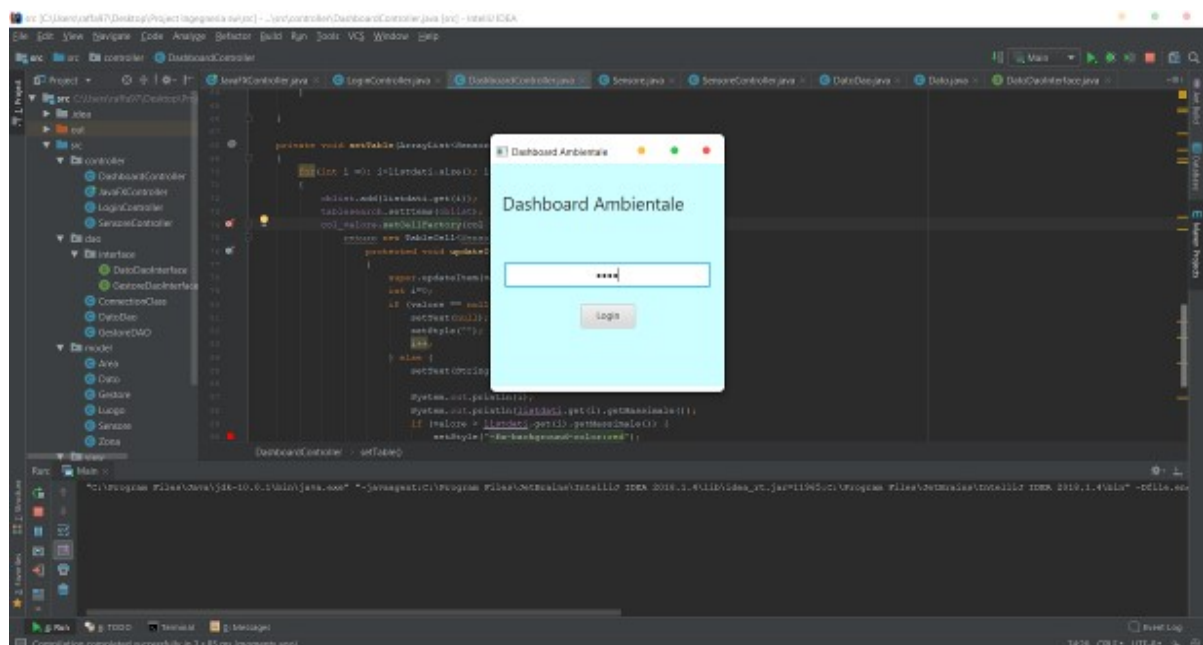


Figura8

Una volta effettuato l'accesso, controllando i dati nel db, il gestore potrà accedere alla dashboard per la visualizzazione dei dati.

A seconda del tipo di gestore verrà caricata una selezione diversa:

Gestore Area:

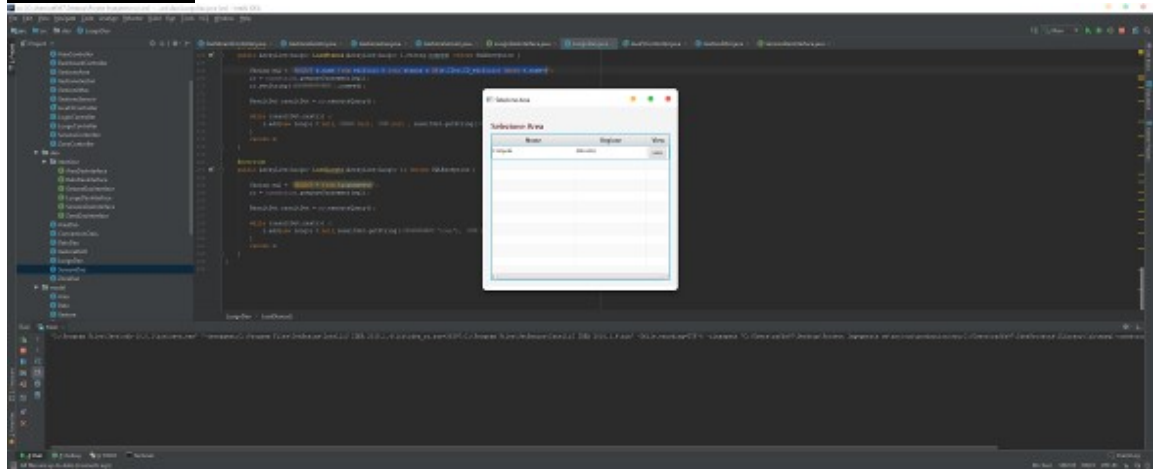


Figura9

Gestore Zona:

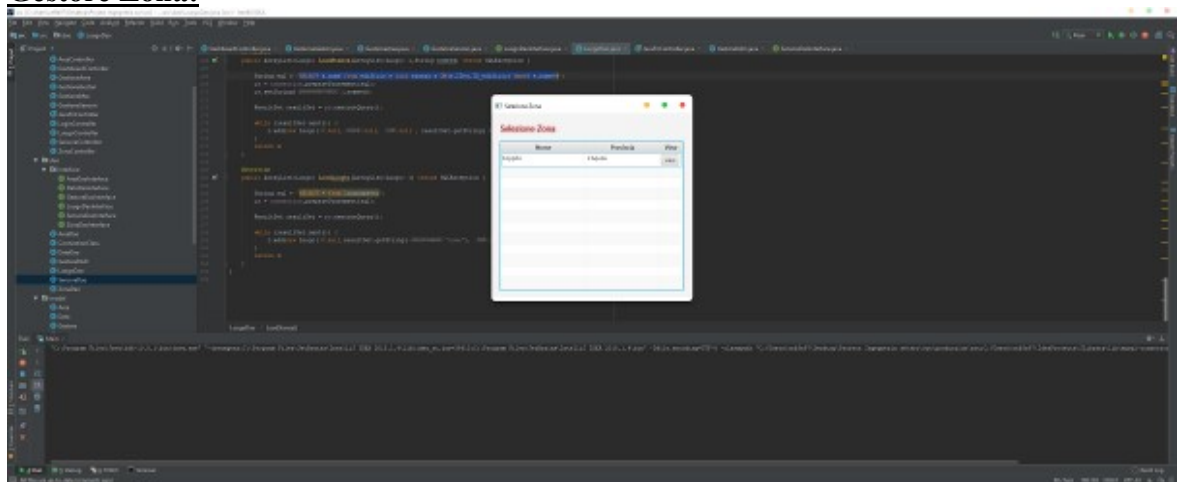
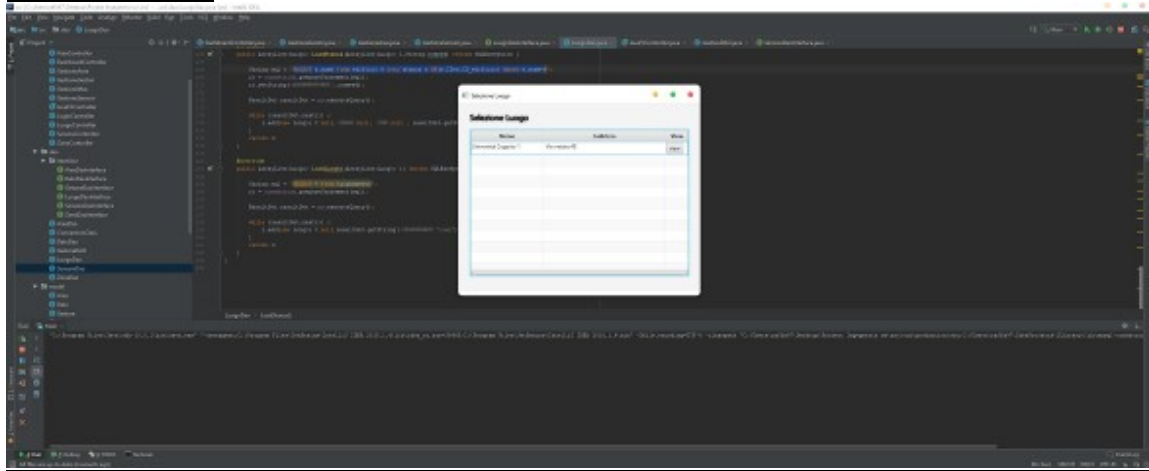


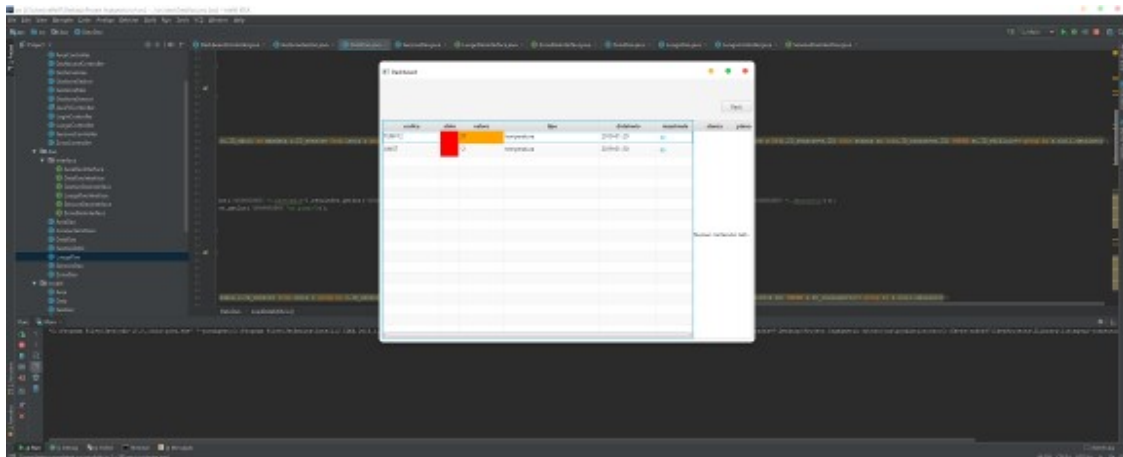
Figura10

Gestore Luogo:Figura11

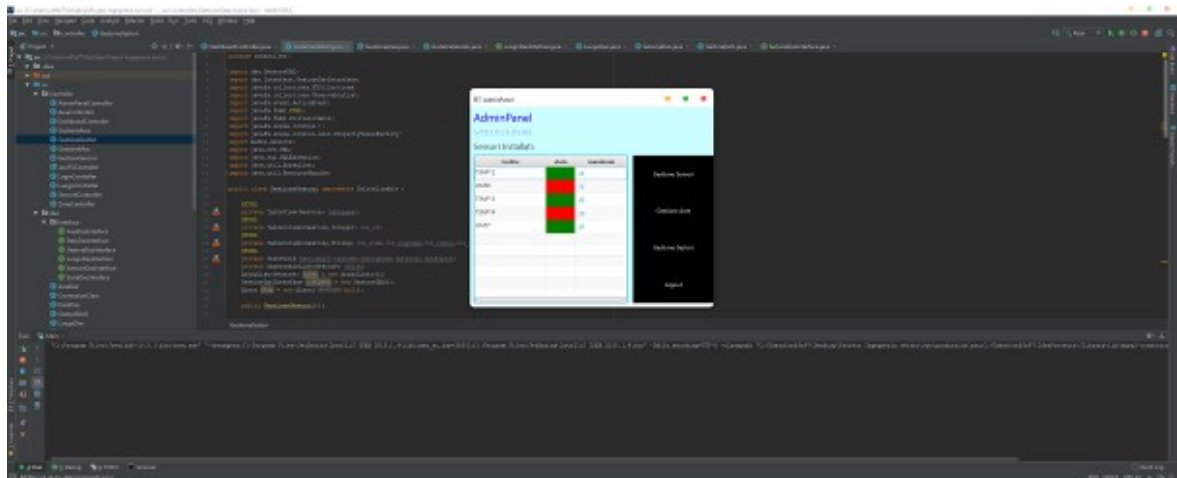
NB: Un gestore dell'area potrà visualizzare tutte le view (area,zona,luogo)

Un gestore della zona potrà visualizzare le view: -zona,luogo.

Dopo la scelta dell'edificio, un qualsiasi tipo di gestore, potrà visualizzare i dati che i sensori acquisiscono minuto per minuto, tramite un apposita dashboard qui sotto riportata:

Figura12

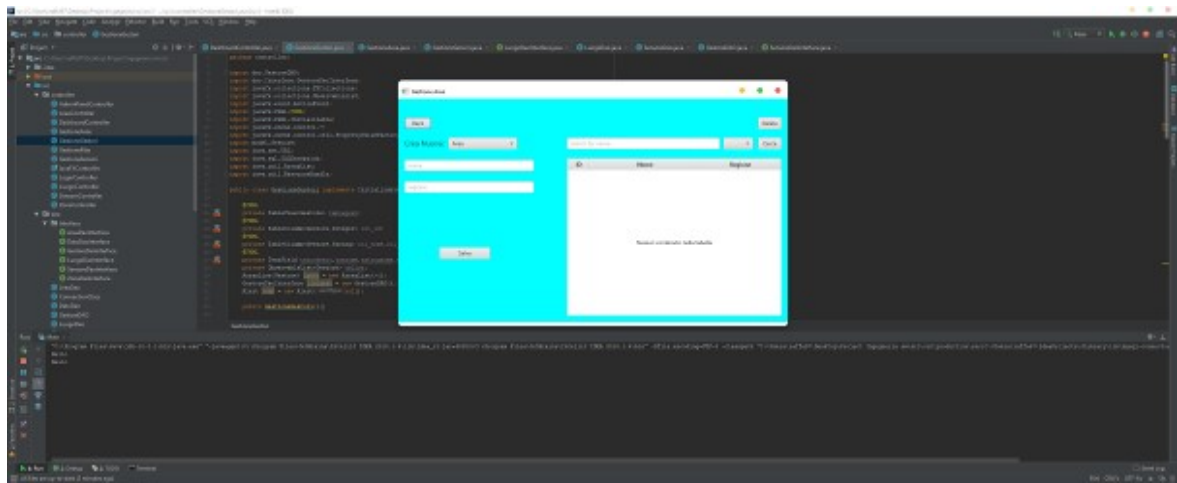
Un admin potrà accedere al suo “adminPanel” per poter gestire in back-end tutto il sistema, dopo l’accesso gli verrà mostrata questa home:



[Figura13](#)

Da questa home, tramite apposito menu laterale, potrò muovermi nelle varie sezioni per andare a gestire le aree, gestori e i sensori.
Qui di seguito verranno riportate le varie grafiche:

GestioneAree:



[Figura14](#)

GestioneGestori:

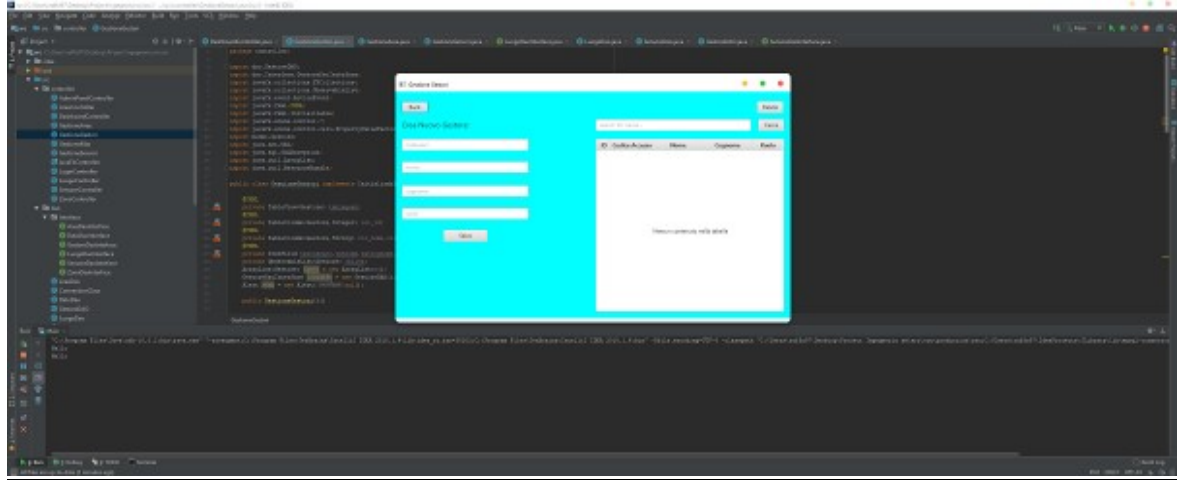


Figura15

GestioneSensori:

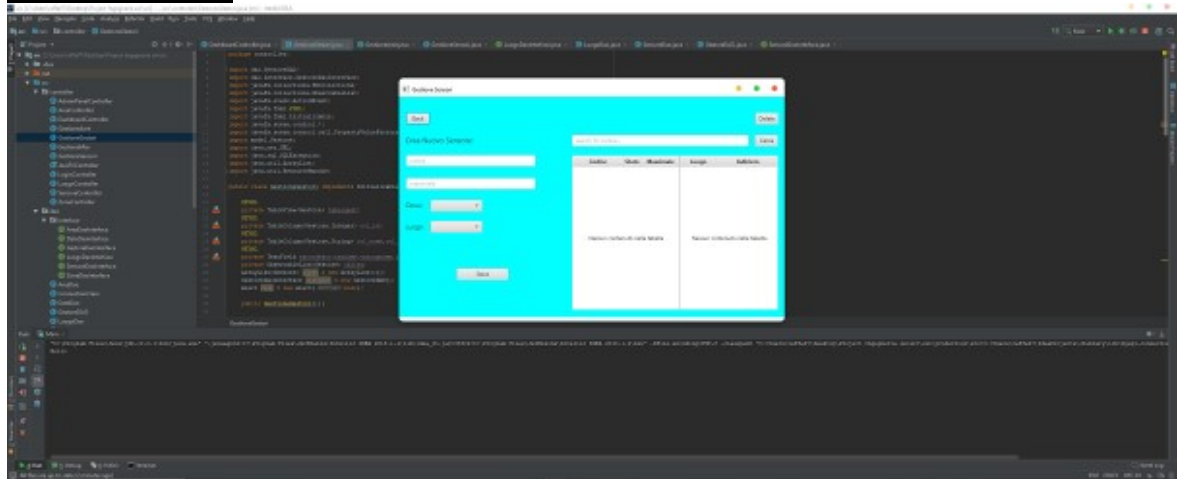


Figura16