

Service Oriented for Software Engineering

SMART HOME SAVES

CRISTINA CIAVARRO 253188

DI NATALE MARCO 255660



Overview

SMART HOME SAVES

Il nostro caso di studio è una Smart Home che ha come goal il risparmio, inteso sia sotto forma di risparmio energetico (energy saving) che come risparmio economico (economic saving). Per realizzare una Smart Home con queste caratteristiche abbiamo analizzato lo stato dell'arte di ecosistemi smart o smart Device che avessero queste caratteristiche, come ad esempio i termostati intelligenti di ultima generazione o smart home già esistenti.

Per quanto riguarda la struttura della casa, abbiamo deciso di implementare l'ecosistema utilizzando tre stanze: cucina, giardino (o più in generale esterno della casa) e bagno. Naturalmente il sistema funziona anche per un numero maggiore di stanze.

La realizzazione del software della Smart Home è stata implementata tramite l'utilizzo di Openhab 2, adottando come paradigma un MAPE K-loop. Illustriamo brevemente come sono state affrontate ed implementate le parti caratterizzanti di questo pattern:

MONITORING : il monitoraggio è stato realizzato tramite l'utilizzo di sensori che rilevano la temperatura, la luminosità e il movimento. I valori sono forniti al sistema grazie a Arduino.

ANALIZE : i dati rilevati dai sensori sono salvati su un database. Tramite degli script e delle query opportune questi dati vengono filtrati per poter effettuare le predizioni.

PLANNING : la parte relativa alla pianificazione delle azioni è implementata nelle rule nella configurazione di Openhab.

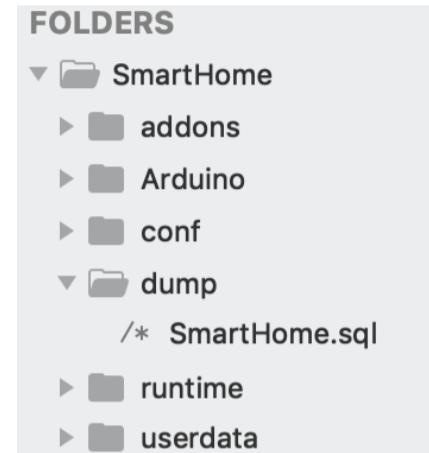
EXECUTED : l'esecuzione delle azioni per gli attuatori è stata realizzata tramite Arduino. Naturalmente non sono stati usati realmente tutti gli attuatori presentati nel progetto (come ad esempio il window shutter o i termosifoni) sia per motivi economici che per

motivi logistici, tuttavia l'esecuzione delle azioni è stata simulata tramite l'accensione e lo spegnimento di led.

KNOWLEDGE : la conoscenza risiede nei database che il monitoring e l'analisi riempiono e modellano. Sono stati inoltre implementati algoritmi di previsione per rendere il sistema più intelligente. In particolare la Smart Home è in grado di imparare le abitudini dell'utente per prevedere il suo ritorno a casa, così da poter adottare misure di risparmio energetico per quanto riguarda la temperatura. Infatti il sistema può anche prevedere le tempistiche impiegate della casa per riscaldare ogni stanza, così da poter iniziare a riscaldare gli ambienti in modo da far trovare all'utente la temperatura desiderata al suo ritorno.

HOW TO

Per poter iniziare a lavorare è necessario prima di tutto importare in phpmyadmin il database creato per questo progetto. Quest'ultimo si trova all'interno di **SmartHome/dump**.



Importato il database è necessario configurare lo script di connessione con phpmyadmin presente in **SmartHome/conf/scripts/config_mysql.py** .

```
1 import mysql.connector
2
3
4 def connection():
5
6     hostName="localhost"
7     portNumber="8889"
8     userName="root"
9     userPasswd="root"
10    databaseName="smarthome"
11
12    return mysql.connector.connect(host=hostName,
13                                    port=portNumber,
14                                    user=userName,
15                                    passwd=userPasswd,
16                                    database=databaseName)
17
```

Stabilita la connessione prima di poter utilizzare gli script in python all'interno del progetto è necessario definire il giusto **PATH** di residenza degli script.

Nella cartella **SmartHome/conf/rules** si trova il file **demo.rules** nel quale sono definite tutte le rules del progetto. All'inizio di questo file è definita una variabile **commonPath** che modella il comando che verrà lanciato da openhab tramite **shell**.

```

1 var String commandPath = "python /Users/xxxxxxxx/Downloads/SmartHome/conf/scripts"
2
3
4
5 rule "Set Default Values"
6 when
7 | | | | System started
8 then
9
10 /* Voglio settare tutte le variabili solo al primo avvio del sistema, quindi faccio un controllo su una variabile
11 | e se questa è uguale ad uno stato indefinito allora il sistema deve essere */
12 if((TemperatureK.state == UNDEF || TemperatureK.state == NULL)&&
13 (TemperatureBT.state == UNDEF || TemperatureBT.state == NULL)){
14
15 ManualMode.postUpdate(OFF)
16 PresenceatHome.postUpdate(OFF)

```

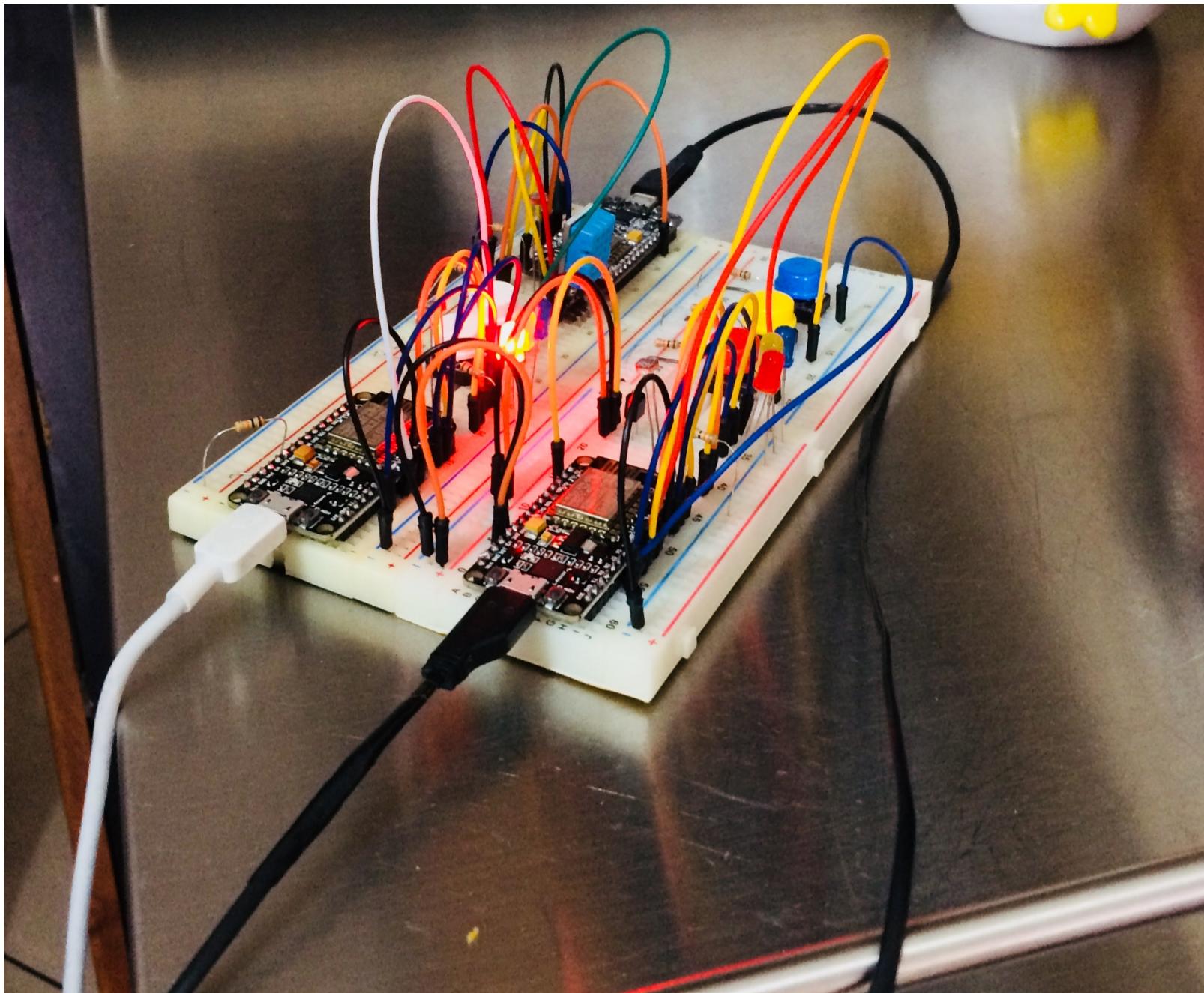
Attenzione: per fare in modo che il comando venga lanciato è necessario avere istallato sul proprio Computer una versione di **Python**.

Ecco un esempio che mostra come mandare in esecuzione uno script esterno:

```

rule "Preference Temperature Kitchen"
when
| Item MinTemperatureK changed or
| Item ThermostatK changed
then
| executeCommandLine(commandPath+"/settings.py Kitchen Temperature "+MinTemperatureK.state+" "+ThermostatK.state)
end

```



Cap 2

MONITORING

I sensori utilizzati sono:

2 sensori di temperatura

3 sensori per rilevare la luminosità

Ognuno di questi è impiegato per monitorare le stanze della Smart Home realizzata. La parte software relativa alle azioni di monitoraggio è stata implementata tramite Arduino, se si desidera visionare il codice è possibile consultarlo nella cartella "Arduino" presente nel progetto inviato.

Per facilitare le azioni di monitoraggio dal punto di vista dell'utente sono stati realizzati anche due pannelli nella sezione Habpanel e Basic UI. Di seguito mostriamo delle immagini come appaiono questi due pannelli:

Habpanel

The screenshot displays the Habpanel interface with several cards:

- Manual Mode:** Shows a lock icon and a power button labeled "OFF".
- External Temperature:** Shows a thermometer icon and the value **26 °C**.
- Thermostat Kitchen:** Shows a thermometer icon and the value **26.0°C**. It includes fields for External Temperature, Internal Temperature (21.1°C), and Min Temperature (0°C). A slider for Ideal Temperature ranges from 0°C to 30°C, with 16.0°C set.
- Thermostat Bathroom:** Shows a thermometer icon and the value **26.0°C**. It includes fields for External Temperature, Internal Temperature (0.0°C), and Min Temperature (0°C). A slider for Ideal Temperature ranges from 0°C to 30°C, with 18.0°C set.
- Mobile Phone:** Shows a battery icon and the value **Mobile Phone**. It includes fields for Presence Mobile User (OFF), Distance From Home (0.0 Km), and Time Outside (0.0 Min). An "Alert Mobile" button is present.
- Presence Kitchen:** Shows a person icon and the value **Presence at home OFF**.
- Shutter Windows:** Shows a window icon and the value **Kitchen OFF**.
- Radiators:** Shows a radiator icon and the value **Kitchen OFF**.
- Light Kitchen:** Shows a lightbulb icon and the value **100.0%**. It includes fields for External Brightness and Internal Brightness, both set to 100.0%. A slider for Min Light ranges from 0% to 100%, with 41.5% set.
- Light Bathroom:** Shows a lightbulb icon and the value **100.0%**. It includes fields for External Brightness and Internal Brightness, both set to 100.0%. A slider for Min Light ranges from 0% to 100%, with 36.5% set.
- Presence Bathroom:** Shows a person icon and the value **Presence Bathroom OFF**.
- Presence Garden:** Shows a person icon and the value **Presence Garden OFF**.

Basic UI :

The screenshot displays the Basic UI interface with several sections:

- Setting:** Includes a "Manual Mode" switch (OFF) and a "Presence at Home" switch (OFF).
- User Phone:** Includes a "Presence mobile User at Home" switch (OFF) and a "Distance User from Home" field (0,0 km). It also includes a "user is coming back home" switch (OFF) and a "user is outside of" field (0,0 min).
- Kitchen:** Includes a "Sensor Motion" switch (OFF), a "Min Temperature" slider (- °C), a "Thermostat" slider (- °C), a "Temperature Sensor" switch (0 °C), a "Light Sensor" switch (0,0 Lux), a "Lamp" switch (OFF), a "Window" switch (OFF), and a "Radiator" switch (OFF).

ANALIZE

La parte di Analisi dei dati comprende la rilettura del database ed il filtraggio di questi. In particolare viene affrontata per preparare i dati per le previsioni della Knowledge che riguardano il rientro dell'utente a casa (quando scorda il cellulare nell'abitazione) e la previsione del tempo impiegato per una stanza a riscaldarsi dato un certo clima interno ed esterno alla Smart Home.

Per comodità spiegheremo insieme la parte di Analize e Monitoring, poiché lavorano in sincronia.

PREVISIONE RIENTRO : il sistema lancia uno script nel momento in cui l'utente esce di casa lasciando la casa vuota, e scorda il cellulare. Questo script controlla l'orario di uscita e calcola una fascia oraria che va da 20 minuti prima a 20 minuti dopo che l'utente è uscito di casa. Avendo presente questa fascia oraria, cerca nel database informazioni relative alle vecchie uscite fatte dall'utente in quell'intervallo di tempo in quello stesso giorno della settimana. Se esistono salva tutti gli orari di ritorno e ne calcola una media che sarà proprio la predizione dell'orario di ritorno dell'utente.

PREVISIONE TEMPERATURA : conoscendo l'orario di ritorno dell'utente (sia tramite predizione che tramite le informazioni fornite dal cellulare), è comodo far predire al sistema il tempo che impiega a riscaldare una stanza, cosicché possa far trovare all'utente la stanza calda al suo ritorno. Sempre tramite uno script, che viene lanciato poco prima che l'utente torni a casa, il sistema controlla la temperatura della stanza in quel momento per selezionare tutte le righe dello storico in cui il sistema aveva:

- una temperatura interna a quella stanza che fosse nel range di un grado rispetto quella in quel momento,
- una temperatura esterna che fosse nel range di un grado di quella in quel momento,
- una temperatura desiderata (temperature goal) che fosse uguale all'ultima impostata dall'utente.

Trovate le righe che fanno match con quanto detto, il sistema estrae i tempi che ha impiegato ogni volta la casa a raggiungere quel temperature goal e ne calcola la media.

Il risultato sarà proprio la predizione del tempo necessario per raggiungere quella temperatura.

PLANNING

La parte di pianificazione è stata implementata nelle rule di Openhab. Illuatriamo ora le regole anche tramite sequence dyagram. Se si preferisce consultare le immagini separatamente o, se si preferisce, l'intero progetto uml, basta accedere alla cartella "uml" del progetto. Dal momento che la smart home ha come goal il risparmio, la pianificazione delle azioni si concentra su due delle azioni che fanno sprecare energia in una casa: il riscaldamento e l'illuminazione.

HEATING : la smart home monitora il riscaldamento a seconda delle abitudini e delle preferenze dell'utente. Infatti se l'utente è in casa (ossia un sensore di movimento rileva la presenza dell'utente in almeno una stanza della casa), viene portata la temperatura di ogni stanza alla temperatura gradita dall'utente. È possibile cambiare questa temperatura anche in tempo reale attraverso il termostato. La casa cercherà di portare ogni stanza alla temperatura gradita dall'utente sfruttando anche la temperatura esterna : infatti grazie ai sensori di temperatura posizionati all'interno di ogni stanza ed all'esterno della casa, il sistema aprirà la finestra per far entrare calore naturale, spegnendo quindi i termosifoni. Naturalmente se la temperatura cambia e l'apertura della finestra inizierà a causare una dispersione di calore piuttosto che un guadagno di quest'ultimo, il sistema chiuderà la finestra e azionerà i termosifoni se necessario.

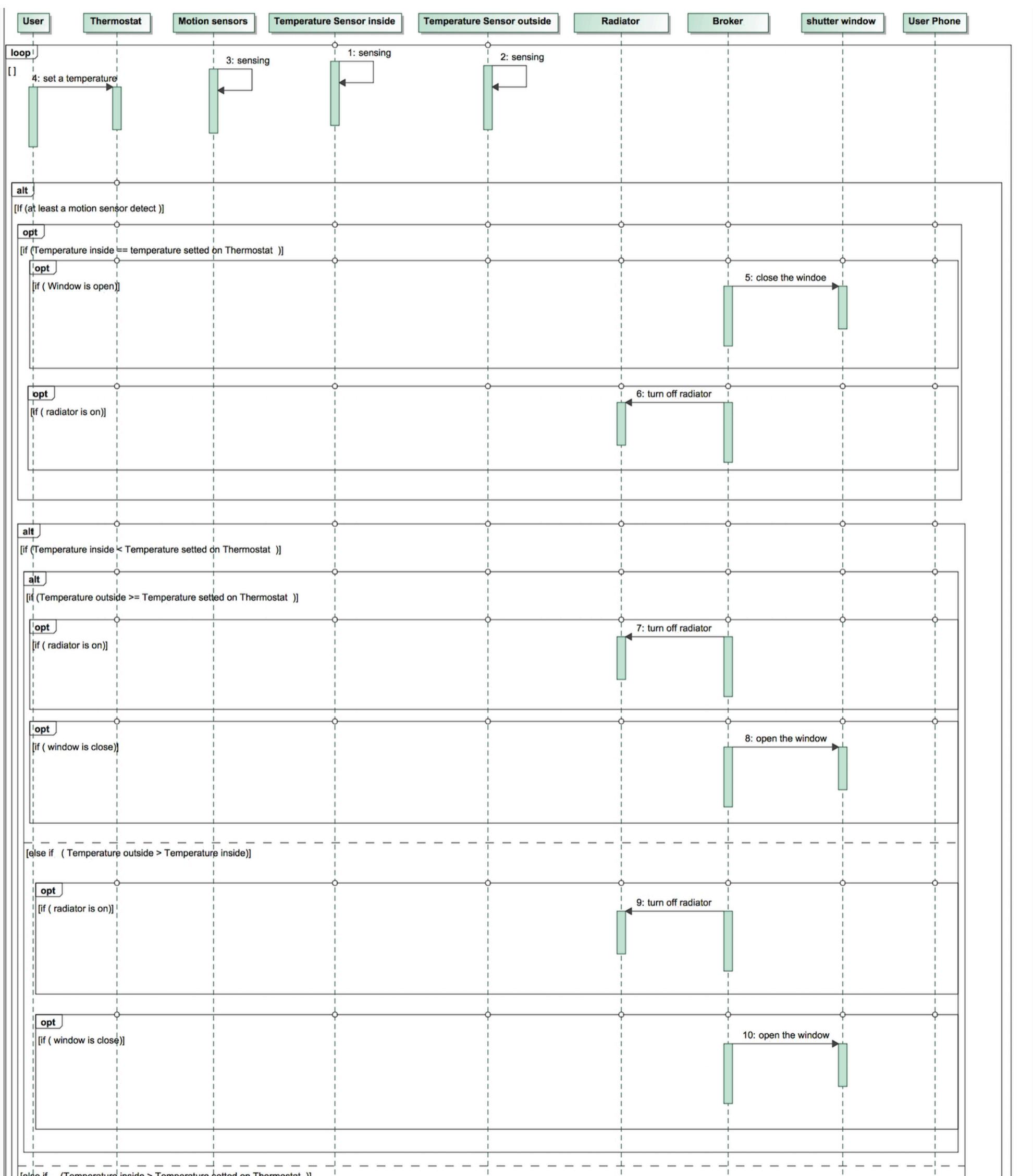
Se l'utente non è in casa (ossia nessun sensore di movimento rileva la presenza di un utente in nessuna stanza della casa) il sistema chiude tutte le finestre per isolare la casa dalla temperatura esterna e dagli estranei.

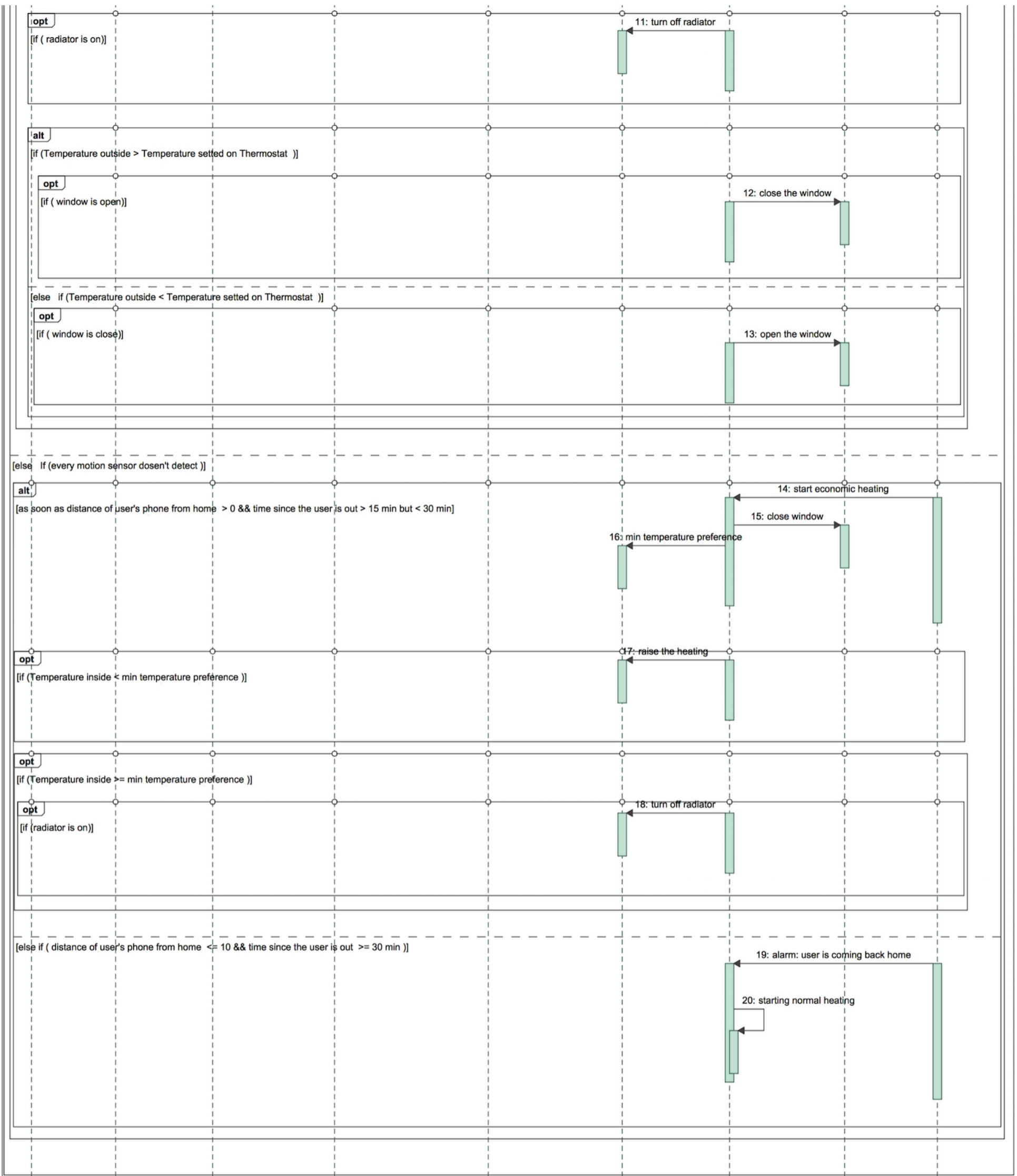
Tuttavia grazie al cellulare dell'utente è possibile anche controllare in tempo reale, quando l'utente esce di casa con il device, il momento in cui sta tornando a casa. Infatti il sistema controlla quando l'utente si allontana da casa di almeno 1 metro e, se è fuori da almeno 10 minuti, considera l'uscita dell'utente un'uscita significativamente lunga.

Quando l'utente esce di casa il sistema spegne i termosifoni e lascia la temperatura delle stanze uguale almeno alla temperatura minima scelta dall'utente. Questa temperatura minima è una preferenza data dall'user al sistema ed è più bassa rispetto a quella scelta dall'utente quando è in casa.

Appena l'utente, passata almeno mezz'ora da quando è uscito di casa, sta tornando verso casa (quindi i km di distanza da casa diminuiscono) il sistema se ne accorge grazie

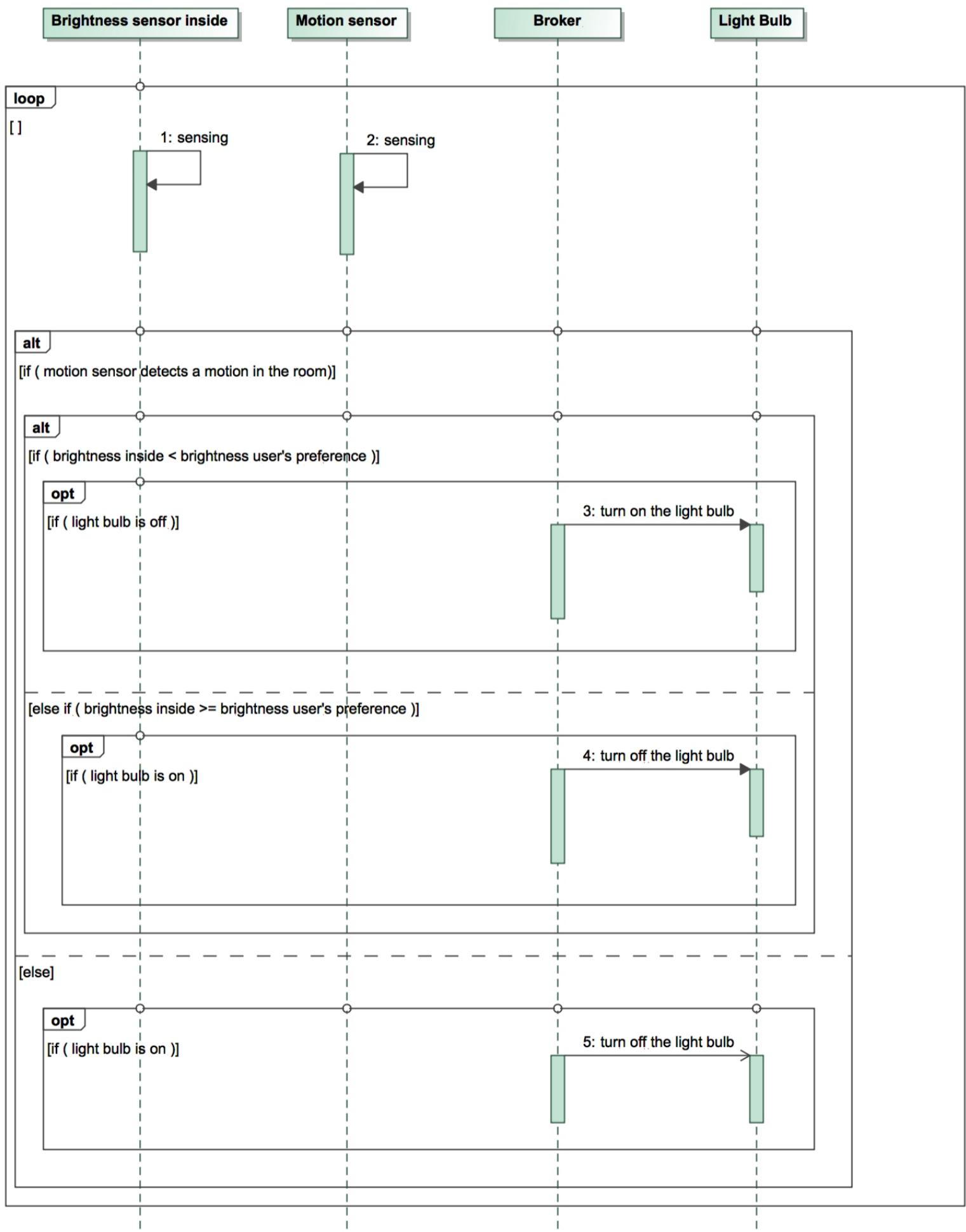
allo spostamento del cellulare, che comunica il ritorno dell'utente. Grazie a questo la Smart Home riporta la temperatura delle stanze alla temperatura gradita dall'utente così da fargli ritrovare le stanze calde al suo ritorno e diminuendo così gli sprechi.





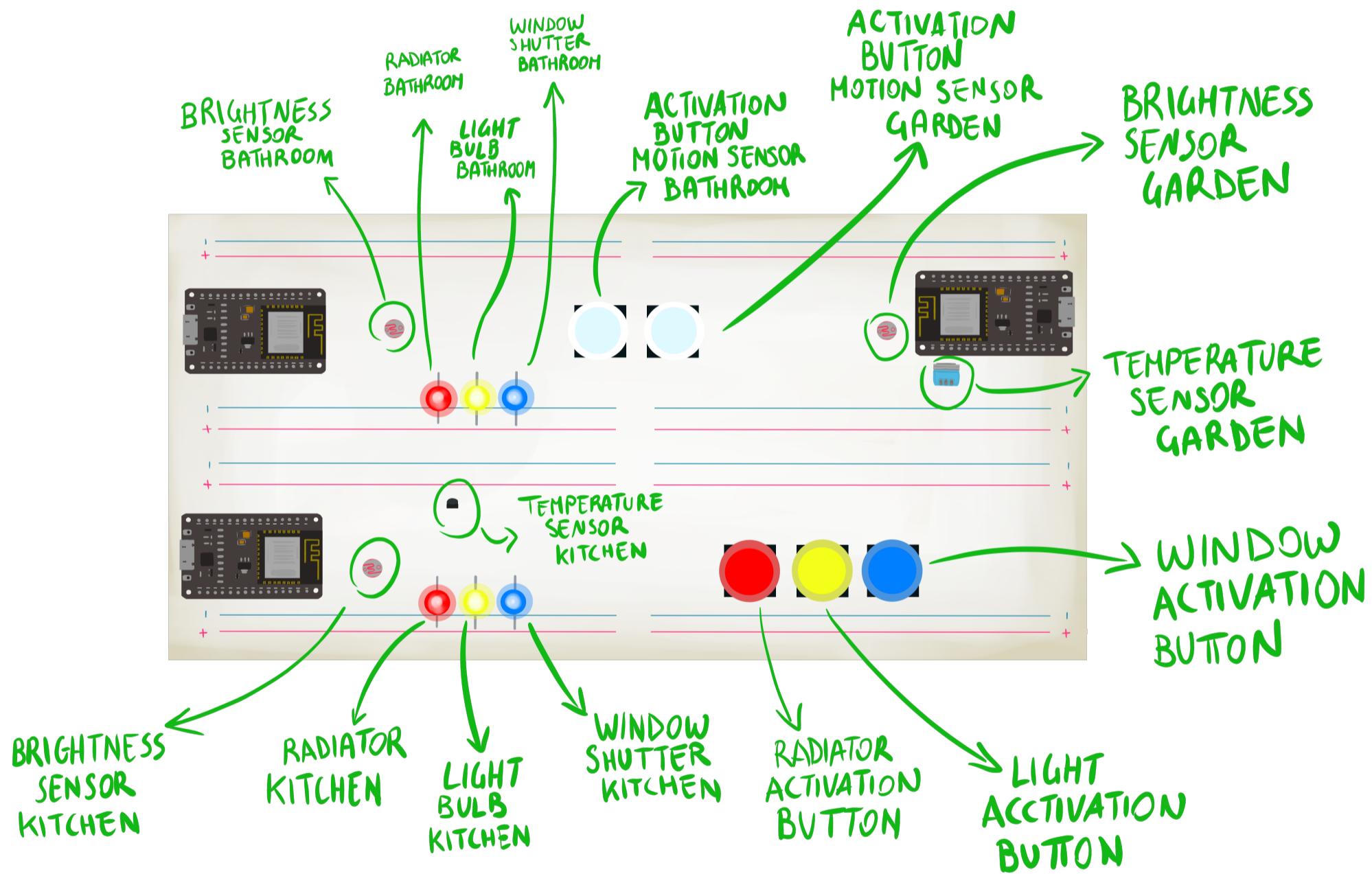
LIGHT : il sistema permette all'utente di impostare una preferenza sulla quantità di luce massima gradita, così da far spegnere le luci all'interno di una stanza quando il sensore di luminosità all'interno di quella stanza rileverà una quantità di luce maggiore di quella

scelta. In questo modo c'è un risparmio di energia. Un ulteriore fattore di risparmio è il fatto che le luci si accendono solo quando il sensore di movimento rileva un movimento in quella stanza e si spengono non appena il sensore si accorge che l'utente ha lasciato la stanza.



EXECUTED

L'esecuzione delle azioni decise dal planning è simulata con l'ausilio di pulsanti e led. L'immagine sottostante aiuterà a comprendere meglio la realizzazione:



Il planning può decidere di eseguire una delle seguenti azioni relative agli attuatori:

- apri/chiudi finestra bagno/cucina (led blu),
- accendi/ spegni la luce bagno/cucina (led giallo),
- accendi/ spegni termosifoni bagno/cucina (led rosso).

Quindi, per esempio, quando il planning deciderà di aprire la finestra in cucina, il led blu della cucina si accenderà; se invece voglio aprire manualmente la finestra della cucina (ad esempio perché ho impostato la Manuale Mode), basterà cliccare il pulsante blu.

L'hardware usato per il progetto comprende:

- 3 schede NodeMCU
- 6 led
- 5 pulsanti
- 1 sensore di temperatura **LM35**
- 1 sensore di temperatura ed umidità **DHT11**
- 3 sensori per il rilevamento della luminosità **Fotoresistenze**

Se si dispone di questi hardware forniamo, insieme al progetto, gli stack di Arduino da poter utilizzare per l'esecuzione completa del progetto. Gli stack sono disponibili nella cartella **SmartHome/Arduino**. In mancanza dell'hardware è possibile scrivere sul Broker MQTT direttamente tramite shell (una volta installato mosquitto sul proprio computer) con il comando:

mosquitto_pub -t NOME_CANALE(State Topic) -m PARAMETRO

Ad esempio se si vuole accendere la lampadina della cucina tramite broker MQTT, basta scrivere sulla shell:

mosquitto_pub -t kitchen/sensor/lamp/state -m ON

Se si vuole ricevere un comando da un item basta digitare il comando:

mosquitto_sub -t NOME_CANALE(Command Topic)

Ad esempio se si vuole ricevere un comando dal termosifone della cucina tramite broker MQTT, basta scrivere sulla shell:

mosquitto_sub -t kitchen/command/radiator/set

Lista dei canali MQTT per la cucina:

| Item | State Topic | Command Topic |
|-----------------|---|------------------------------------|
| MotionK | kitchen/sensor/motion/state | |
| ThermostatK | kitchen/sensor/thermostat/state | kitchen/command/thermostat/set |
| MinTemperatureK | kitchen/preference/minTemperature/state | kitchen/command/minTemperature/set |
| TemperatureK | kitchen/sensor/temperature/state | |
| MinLightK | kitchen/preference/minLight/state | kitchen/command/minLight/set |
| lightK | kitchen/sensor/light/state | |
| lampK | kitchen/sensor/lamp/state | kitchen/command/lamp/set |
| shutterK | kitchen/sensor/shutter/state | kitchen/command/shutter/set |
| radiatorK | kitchen/sensor/radiator/state | kitchen/command/radiator/set |

Lista dei canali MQTT per il bagno:

| Item | State Topic | Command Topic |
|------------------|---|--------------------------------------|
| MotionK | bathroom/sensor/motion/state | |
| ThermostatBT | bathroom /sensor/thermostat/state | bathroom /command/thermostat/set |
| MinTemperatureBT | bathroom /preference/minTemperature/state | bathroom /command/minTemperature/set |
| TemperatureBT | bathroom /sensor/temperature/state | |
| MinLightBT | bathroom /preference/minLight/state | bathroom /command/minLight/set |
| lightBT | bathroom /sensor/light/state | |
| lampBT | bathroom /sensor/lamp/state | bathroom /command/lamp/set |
| shutterBT | bathroom /sensor/shutter/state | bathroom /command/shutter/set |
| radiatorBT | bathroom /sensor/radiator/state | bathroom /command/radiator/set |

Lista dei canali MQTT per il giardino:

| Item | State Topic | Command Topic |
|---------------|----------------------------------|----------------------|
| MotionK | garden/sensor/motion/state | |
| TemperatureBT | garden /sensor/temperature/state | |
| lightBT | garden /sensor/light/state | |

Lista dei canali MQTT per il cellulare:

| Item | State Topic | Command Topic |
|--------------------|-------------------------|----------------------------|
| PresenceMobileUser | phone/presence/state | phone/presence/command/set |
| DistanceFromHome | phone/distance/state | |
| alertMobile | phone/alert/state | phone/alert/command/set |
| TimeOutside | phone/timeoutside/state | |

Lista dei canali MQTT per le impostazioni:

| Item | State Topic | Command Topic |
|----------------|-----------------------------|---------------------------------|
| ManualMode | settings/manualmode/state | settings/manualmode/command/set |
| PresenceatHome | settings/presencehome/state | settings/presencehome/set |

KNOWLEDGE

Per adattare il sistema alla casa e agli utenti abbiamo definito delle rules che invocano degli script scritti in python e che ci permettono di far adattare la smart home al comportamento degli utenti. Gli script che descriveremo di seguito sono invocati mediante il comando di openhab **executeCommandLine()**.

| Script | Parameters | Description |
|--|---|---|
| config_mysql.py | host, port, user, passwd, database | Stabilisce la connessione con il Database Mysql |
| settings.py | room, sensor, min_temperature, ideal_temperature | Ogni volta che l'utente cambia una preferenza relativa alla luce o al riscaldamento (MinTemperatureK, MinTemperatureBT, ThermostatK, ThermostatBT, MinLightK, MinLightBT), viene invocato lo script che provvede a salvare sul database la nuova preferenza dell'utente. Lo script prende in input 4 parametri: <ol style="list-style-type: none">1. il primo identifica la stanza per il quale si sta memorizzando la preferenza,2. il secondo identifica per quale tipologia di apparato si sta memorizzando la preferenza (Temperature o Light)3. Il terzo la minima temperatura (o luce) che si vuole avere nella stanza4. L'ultimo la massima temperatura (o luce) che si vuole avere nella stanza. |

getSettings.pyroom,
sensor,
type

All'avvio del sistema, per ripristinare il settaggio dei parametri definiti dall'utente, il sistema invocherà per ogni tipologia di apparato lo script passando come input:

1. la stanza per il quale si vuole ottenere la preferenza,
2. l'apparato (sensore)
3. il tipo di preferenza che si vuole recuperare.

Per esempio questo script viene usato per il recupero della minima temperatura che si vuole avere nella stanza.

homeEmpty.py

room

Appena il sistema non percepisce più movimenti all'interno di una stanza, registra sul database l'avvenimento segnando l'ora, il giorno della settimana e la stanza. Se in nessuna stanza viene più rilevato un movimento il sistema registrerà come stanza vuota l'intera casa

fullHome.py

room

Quando il sistema torna a percepire un movimento in una stanza viene lanciato lo script che verifica la data in cui l'utente era uscito da quella stanza. Se il tempo di assenza dalla stanza è significativo (cioè se è maggiore di 30 minuti) il sistema aggiornerà l'informazione in precedenza salvata memorizzando la data di rientro e di conseguenza la fascia oraria in cui l'utente era uscito. Nel caso in cui l'utente rientra nella stanza in un lasso di tempo inferiore a 30 minuti, reputiamo l'informazione poco utile e di conseguenza, il dato in precedenza memorizzato, verrà eliminato.

startingHeating.pyroom,
external_temperature,
internal_temperature,
goal_degrees

Ogni volta che il Sistema provvede ad avviare l'impianto di riscaldamento viene memorizzata l'ora in cui si è avviato, la temperatura interna e quella esterna alla stanza e la temperatura che si vuole raggiungere in quella stanza.

endingHeating.py

room

Raggiunta la temperatura desiderata dall'utente all'interno di una stanza, il sistema andrà ad aggiornare l'informazione memorizzata all'avvio dell'impianto di riscaldamento e memorizzerà l'ora in cui la stanza è arrivata al questa temperatura.

In questo modo siamo in grado di determinare il tempo impiegato dai termosifoni per scaldare la stanza.

estimatedReturnTime.py

input_room,
input_externalT
,input_internalT,
input_goal

Questo script ha come obiettivo quello di stimare il rientro in una stanza (o a casa) dell'utente. Il sistema, avendo come informazioni preliminari il tempo impiegato da ogni singola stanza a raggiungere la temperatura ideale e avendo memorizzato tutti i tempi in cui la casa resta priva di utenti, potrà stimare per ogni giorno della settimana l'ora ipotetica in cui tornerà l'utente. Questo perché il sistema ha memorizzato nei giorni precedenti l'orario di uscita e di rientro dell'utente.

Come detto in precedenza, quando l'utente esce di casa (e non vi è più nessuno al suo interno) il sistema registra quando inizia il momento di inattività al suo interno. Avendo come storico le settimane trascorse, lo script andrà a recuperare tutte le informazioni memorizzate in cui l'utente è uscito nella stessa fascia oraria di quella memorizzata e di cui non si conosce l'ora di rientro. Vedendo la fascia oraria di rientro dei giorni in cui si hanno informazioni, il sistema stima l'ora di rientro per quella giornata. Recuperato il tempo che ci impiega la stanza più lenta a scaldarsi e stimata l'ora di rientro in casa dell'utente, il sistema provvedere ad accendere il riscaldamento nell'ora stimata di rientro meno il tempo impiegato dalla stanza più lenta a riscaldarsi. In questo modo l'utente trova casa riscaldata al suo rientro.