Analisi dei requisiti e Progettazione di un Software per la gestione di una casa domotizzata

Salvatore Maria Mobilia matricola X81000882

Indice

Analisi dei requisiti....3
Scopo....3
Vincoli....3
Ambienti....3
Dispositivi....4
Requisiti generali....4
Operazioni sui dispositivi....5
Menù principale....6
Progettazione Concettuale....8
Progettazione Logica....11
Classi....11
Luce (Light)....11
Condizionatore (Heater)....11
Televisore (Tv)....11
Macchinetta Caffè (Coffeemachine)....12
Pompa (Pump)....12
Spruzzo (Spray)....12
Sistema di irrigazione (IrrigationSystem)....13
Porta Automatica (AutomaticDoor)....13
Giardino anteriore (FrontGarden)....13
Entrata (Entrance)....14
Salotto (Lounge)....14
Bagno (Bathroom)....15
Camera da letto matrimoniale(Doubleroom)....15
Cameretta (Bedroom)....16
Camera degli ospiti (Guestroom)....16
Sala da pranzo (Diningroom)....17
Garage....17
Giardino posteriore (BackGarden)....17
Telecomando (Remote)....18
Diagrammi delle attività....19
Diagrammi di sequenza....20
Scenario 1 : accensione luci garage....21
Scenario 2 : uscita immediata....21

Analisi dei requisiti

Scopo

Lo scopo del software è la gestione della domotica di una casa.

Vincoli

Il sistema viene realizzato come installazione unica . Tutte le operazioni verranno effettuate da un telecomando universale per cui non vi è necessità di distinguere i diversi utilizzatori.

Ambienti

La casa è così composta:

- Giardino anteriore;

- Ingresso;

- Salotto;

- Camera da letto matrimoniale;

- Cameretta;

- Camera ospiti;

- Bagno;

- Sala da pranzo/cucina;

- Garage;

- Giardino posteriore;

Dispositivi

All’interno dei vari ambienti si possono trovare i seguenti dispositivi:

- Luci;

- Condizionatori;

- Televisori;

- Macchinette per il caffè;

- Sistemi di irrigazione(formati da pompe e spruzzi);

- Porte e cancelli elettrici;

Requisiti generali

Il software deve gestire diversi ambienti della casa, attraverso un menù principale dal quale si possono scegliere le varie azioni da compiere.

Il sistema deve attuare una sola azione alla volta, attraverso la selezione apposita dell’opzione.

Il menù principale deve presentare tante voci quante sono le stanze(gli ambienti) da gestire, ed una voce aggiuntiva per i controlli generalizzati.

Per ognuna delle voci del menù principale sono previste tante sottovoci quante sono le azioni che è possibile compiere.

Per controlli generalizzati si intende:

- il settaggio e/o l'accensione e/o lo spegnimento di tutte le luci;

- l'apertura e/o la chiusura di tutte le porte;

- il settaggio e/o l'accensione e/o lo spegnimento di tutti i condizionatori;

- il settaggio e/o l'accensione e/o lo spegnimento di tutti i sistemi di irrigazione;

- la visualizzazione dello stato di tutti i dispositivi.

Vi sono diversi dispositivi. Tutti i dispositivi di un determinato gruppo avranno le stesse caratteristiche per cui , anche se in ambienti diversi , avranno le stesse funzionalità.

In ogni ambiente vi sono più dispositivi appartenenti a gruppi diversi. Di seguito verranno descritte le varie azioni che i dispositivi dovranno poter compiere.

L'opzione "Esci" dovrà essere presente sia nel menù principale che nelle varie sottovoci. Essa permetterà di terminare il programma(menù generale) o di tornare indietro alla schermata di selezione precedente(sottovoci).

Inoltre, per ogni sottovoce del menù generale (esclusa "Esci"), sarà presente una opzione "Check", che permetterà di visualizzare lo stato dei dispositivi del dato ambiente preso in esame.

Operazioni sui dispositivi

- Luci:

    → Accendi / Spegni luce;

    → Regola luminosità (da 1 a 5);

    → Check dello stato.

- Condizionatori:

    → Accendi / Spegni condizionatore;

    → Setta modalità(sole → ID=1,ghiaccio → ID=2,turbo → ID=3,deumidificatore → ID=4);

    → Imposta temperatura (da 18°C a 30°C);

    → Check dello stato.

- Televisori:

    → Accendi / Spegni televisore;

    → Imposta canale (da 1 a 100);

    → Imposta volume (da 0 a 100);

    → Check dello stato.

- Macchinette per il caffè:
    → Accendi / Spegni macchinetta;
    → Prepara Espresso;
    → Prepara Caffè Lungo;
    → Prepara Cappuccino;
    → Check dello stato.
- Sistemi di irrigazione:
    → Accendi / Spegni pompa;
    → Attiva / Disattiva spruzzo;
    → Attiva / Disattiva tutti gli spruzzi;
    → Regola potenza di flusso della pompa;
    → Check dello stato.
- Pompe:
    → Accendi / Spegni pompa;
    → Regola potenza del flusso;
    → Check dello stato.
- Spruzzi:
    → Attiva / Disattiva spruzzo;
    → Check dello stato.
- Porte e cancelli elettrici:
    → Apri / Chiudi porta;
    → Check dello stato.

Menù principale

All’interno del menù principale sarà possibile scegliere una delle seguenti voci:

- Giardino anteriore;
- Ingresso;
- Salotto;
- Camera da letto matrimoniale;
- Cameretta;
- Camera ospiti;
- Bagno;
- Sala da pranzo/cucina;

- Garage;

- Giardino posteriore;

- Generale.

- Esci;

Passiamo ora alla disamina la dislocazione dei vari dispositivi all’interno degli ambienti sopra elencati.

Giardino anteriore:

- N. 1 Cancello elettrico;

- N. 12 Luci;

- N.3 Impianto di irrigazione(5 spruzzi il primo, 8 spruzzi il secondo, 3 spruzzi il terzo + 1 pompa ciascuno).

Ingresso:

- N. 1 Porta elettrica;

- N. 4 Luci.

Salotto:

- N. 8 Luci;

- N. 1 Condizionatore;

- N. 1 Televisore.

Camera da letto matrimoniale:

- N. 4 Luci;

- N. 1 Condizionatore;

- N. 1 Televisore;

Cameretta:

- N. 4 Luci;

- N. 1 Condizionatore;

- N. 1 Televisore;

Camera ospiti:

- N. 4 Luci;

- N. 1 Condizionatore;

- N. 1 Televisore;

Bagno:

- N. 4 Luci;

Sala da pranzo/cucina:

- N. 6 Luci;
- N. 1 Condizionatore;
- N. 1 Macchinetta caffè;
- N. 1 Televisore;

Garage:

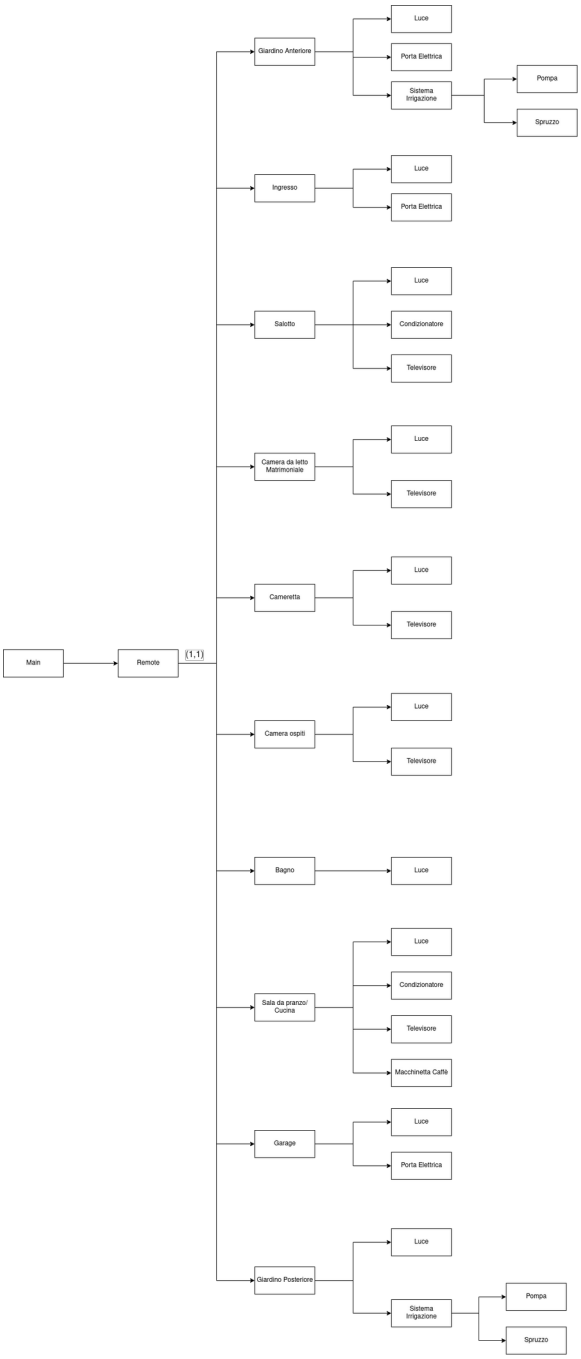
- N. 1 Porta elettrica;
- N. 4 Luci;

Giardino posteriore:

- N. 8 Luci;
- N. 2 Sistemi di irrigazione(4 spruzzi e una pompa ciascuno);

Progettazione Concettuale

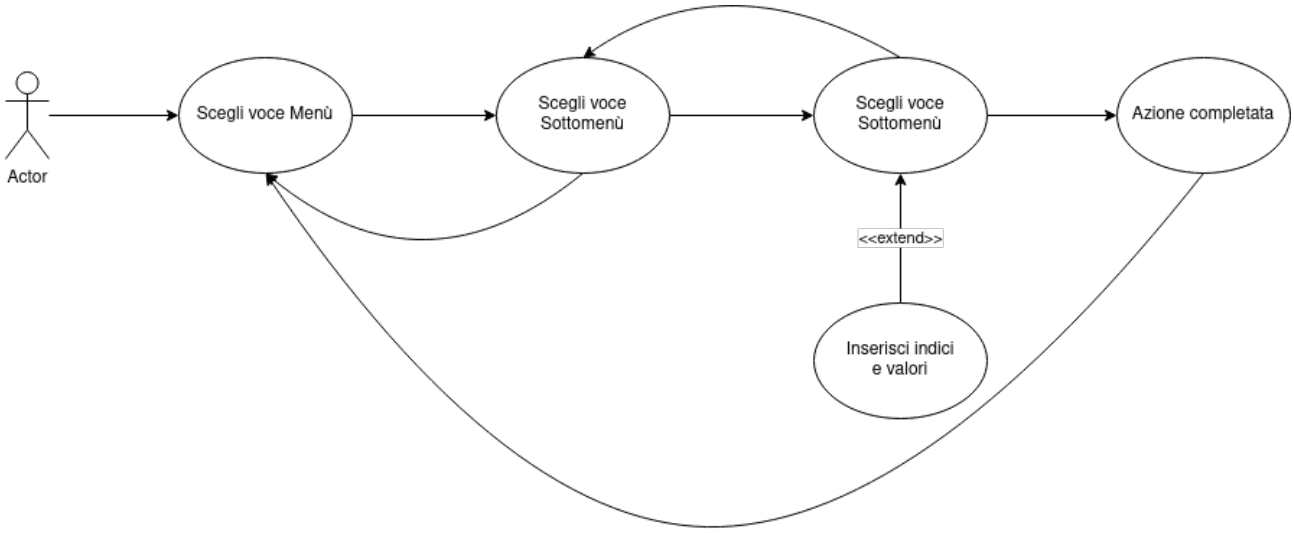
In fase di analisi dei requisiti , il committente ha espressamente richiesto che vi sia un controller che gestisca tutti i dispositivi dell'abitazione. Si è pensato , dunque , di utilizzare un approccio mirato al problema, progettando il software attraverso il Design Pattern Facade. Questo Design Pattern prevede una classe con cui il client "dialogherà" e che andrà a gestire tutte le componenti , in modo tale da "nascondere" al client la reale implementazione e complessità del sistema. Un approccio a blocchi è , infatti , il più adatto per le specifiche fornite. La suddivisione in ambienti suggerisce un multi-layer. Il controller avrà il controllo degli ambienti, che gestiranno a loro volta i dispositivi.



Ecco una prima possibile visione d'insieme dei componenti. Il client si interfaccerà solo con la classe remote, al cui interno verranno create delle istanze delle classi Ambienti(Entrata,Garage ,ecc) descritte in fase di analisi dei requisiti.

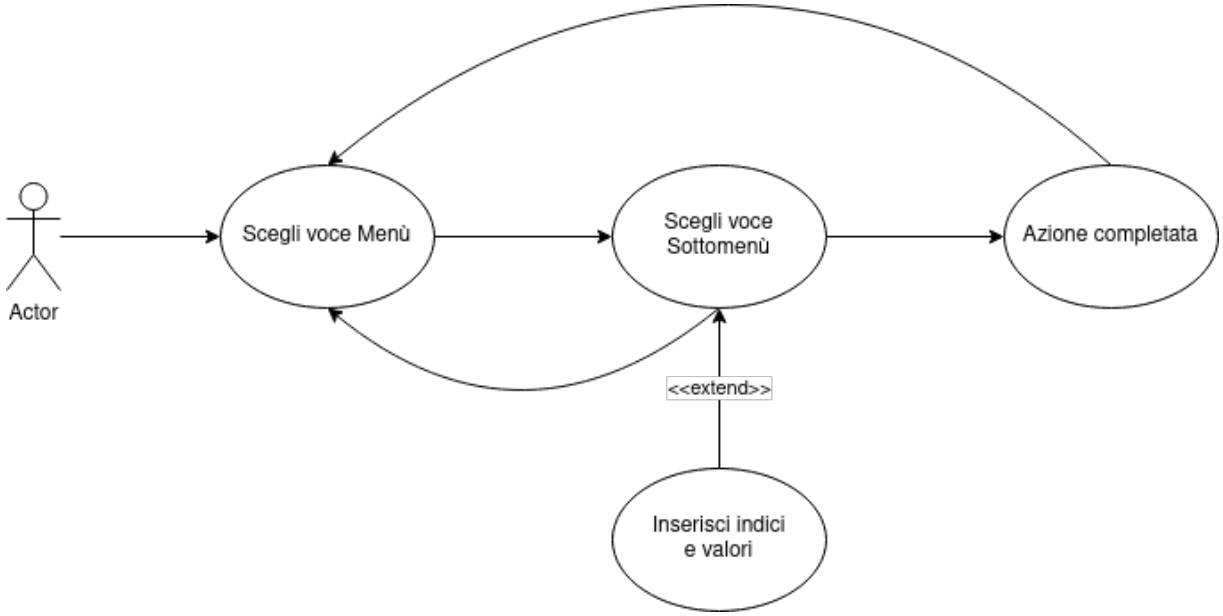
Il modello pensato per l'implementazione del codice è un modello event-driven: l'utente, attraverso l'immissione di valori in input, "scatenerà" l'esecuzione delle varie procedure.

L'analisi dei requisiti ha portato alla luce anche alcune precise caratteristiche di funzionalità del sofware. Per cui si possono già stilare dei diagrammi di casi d'uso. In realtà , essendo il software abbastanza semplice e lineare, esiste una generalizzazione di tutti i casi d'uso presente:

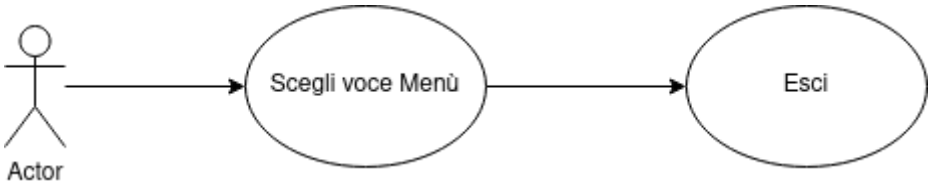


L'attore (utente dell'applicativo) si trova di fronte il menù principale(descritto in seguito). Da questo egli potrà scegliere una voce (ambienti) che lo porteranno ad un sottomenù con altre voci. Da qui egli può scegliere l'azione da far compiere all'applicativo. Alcune di esse necessitano di valori in input , altre no. Ad azione completata, l'attore si troverà nuovamente di fronte al menù principale. In caso di immissione di valori errati(vi sono alcuni controlli all'interno dei metodi) dall'utente, la sequenzialità delle azioni verrà interrotta e si passerà nuovamente al menù principale.

Esistono due variazioni a questo schema generale. La prima prevede la scelta dal menù principale della voce "generale", ossia quella dei controlli generalizzati. In essa si salterà uno step, poiché non vi sarà la necessità di scegliere un ambiente e , di conseguenza , si avrà la scelta dell'azione da far compiere all'applicativo(alcune necessitano di immissione di valori dall'utente). Ecco lo schema:



L’ultima possibile variazione è quella dell’uscita immediata(o dopo aver eseguito azioni) dal programma:



Da questi diagrammi dei casi d’uso sopra mostrati, si evidenzieranno , nella sezione della progettazione logica , i complementari diagrammi delle attività.

Progettazione Logica

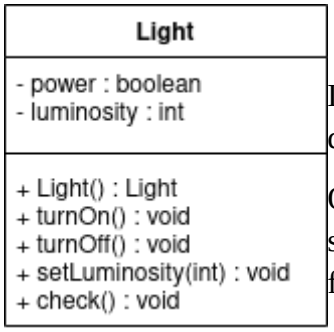
Per soddisfare i requisiti forniti dal committente , si è pensato di attuare un approccio OOP al problema, creando un software in linguaggio Java.

Classi

Analizziamo nel dettaglio le varie classi coinvolte nel processo implementativo.

Luce (Light)

Ecco il diagramma UML della classe Light:

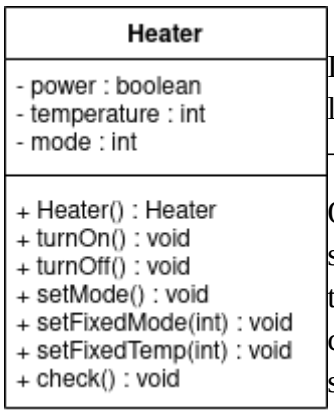


La classe presenta due attributi, un intero ed un booleano, che descrivono lo stato della luce (accesa/spenta – intensità luminosità).

Oltre al costruttore di classe, sono presenti i metodi per accendere / spegnere la luce , settarne la luminosità e per verificare il suo stato di funzionamento.

Condizionatore (Heater)

Ecco il diagramma UML della classe Heater:

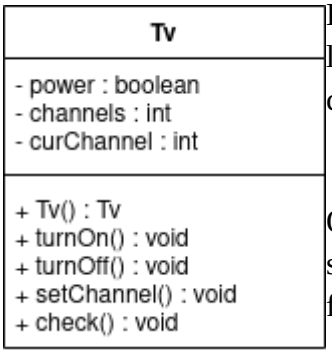


La classe presenta tre attributi, due interi ed un booleano, che descrivono lo stato e i valori interni del condizionatore (acceso/spento – temperatura – modalità).

Oltre al costruttore di classe sono presenti i metodi per: accendere e spegnere il condizionatore , settare il valore di temperatura e modalità tramite parametro attuale , settare il valore di temperatura e modalità chiedendo all’utente l’inserimento e un ultimo metodo per verificare il suo stato di funzionamento ed i valori interni.

Televisore (Tv)

Ecco il diagramma UML della classe Tv:

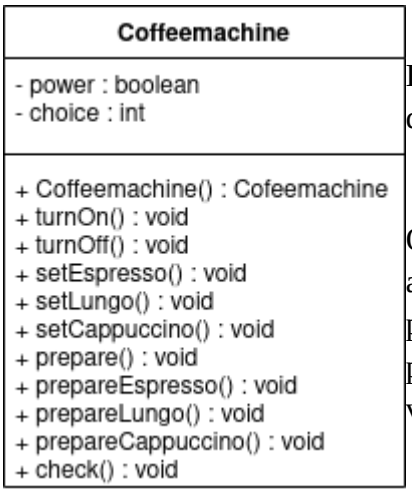


La classe presenta tre attributi, due interi ed un booleano, che descrivono lo stato e i valori interni del televisore (acceso/spento – numero canali – canale corrente)

Oltre al costruttore di classe sono presenti i metodi per accendere e spegnere il televisore , impostare il canale e verificare il suo stato di funzionamento ed i valori interni.

Macchinetta Caffè (Coffeemachine)

Ecco il fiagramma UML della classe Coffeemachine:

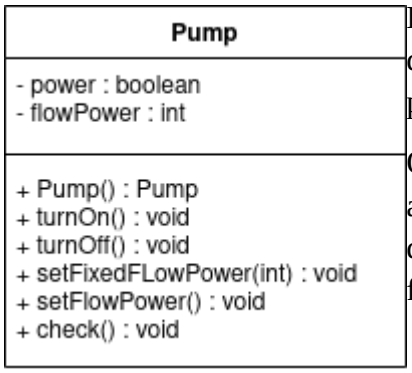


La classe presenta due attributi , uno intero ed uno booleano , che descrivono lo stato (acceso/spento) e la scelta del prodotto.

Oltre al costruttore di classe, la classe presenta i metodi per : accendere / spegnere il device , settare il device per la preparazione dei tre prodotti disponibili , preparare uno dei tre prodotti disponibili e verificare il suo stato di funzionamento ed i valori interni

Pompa (Pump)

Ecco il diagramma UML della classe Pump:

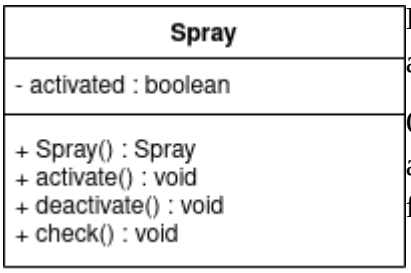


La classe presenta due attributi , un intero ed un booleano , che descrivono lo stato (acceso/spento) e la potenza del flusso della pompa

Oltre al costruttore di classe, la classe presenta i metodi per : accendere / spegnere la pompa , fissare tramite parametro il valore del flusso , settare tramite richiesta testuale all'utente del valore del flusso e verificare il suo stato di funzionamento ed i valori interni.

Spruzzo (Spray)

Ecco il diagramma UML della classe Spray:

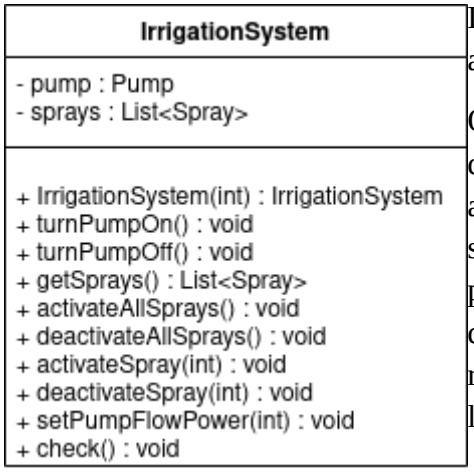


La classe presenta un solo attributo booleano che indica lo stato di attivazione dello spruzzo.

Oltre al costruttore di classe , la classe presenta i metodi per attivare/disattivare lo spruzzo e verificare lo stato di funzionamento.

Sistema di irrigazione (IrrigationSystem)

Ecco il diagramma UML della classe IrrigationSystem:

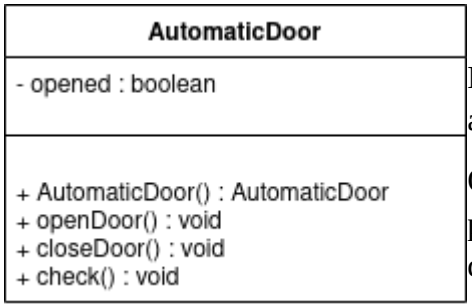


La classe presenta una Pompa e una lista di spruzzi come attributi interni.

Oltre al costruttore(prevede un parametro intero per il numero di spruzzi del sistema) la classe presenta i metodi per : accendere/spegnere la pompa, attivare/disattivare tutti gli spruzzi , attivare / disattivare solo alcuni spruzzi(prevede un parametro intero che indica quale spruzzo attivare o disattivare) , settare la potenza di flusso della pompa , restituire la lista degli spruzzi presenti nel sistema e verificare lo stato di funzionamento dei componenti interni del sistema.

Porta Automatica (AutomaticDoor)

Ecco il diagramma UML della classe AutomaticDoor:

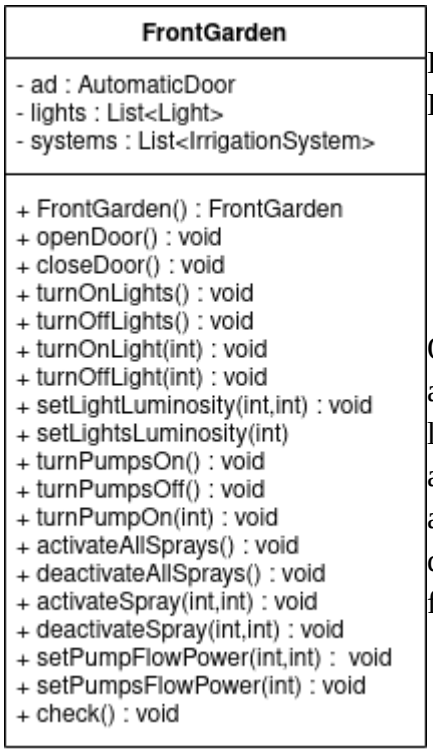


La classe presenta un solo attributo booleano che indica la apertura/chiusura della porta.

Oltre al costruttore , la classe presenta i metodi per aprire la porta , chiudere la porta e per verificare il funzionamento del dispositivo.

Giardino anteriore (FrontGarden)

Ecco il diagramma UML della classe FrontGarden:

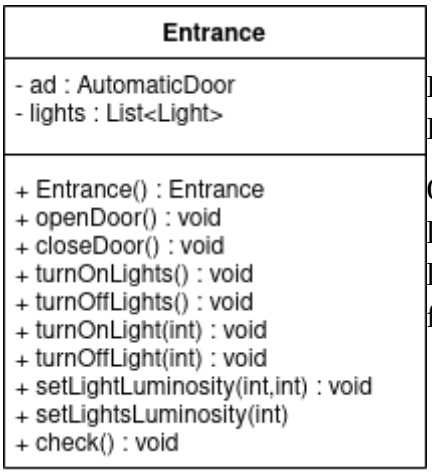


La classe presenta un oggetto AutomaticDoor,una lista di oggetti Light e una lista di oggetti IrigationSystem.

Oltre al costruttore, la classe presenta i metodi per : aprire/chiudere la porta automatica,accendere/spegnere le luci o la singola luce, settare la luminosità di tutte o solo di alcune luci, accendere tutte o solo alcune pompe degli impianti di irrigazione, attivare / disattivare tutti o solo alcuni spruzzi, settare la potenza del flusso di tutte o solo di alcune pompe e verificare lo stato di funzionamento dei vari dispositivi presenti.

Entrata (Entrance)

Ecco il diagramma UML della classe Entrance:

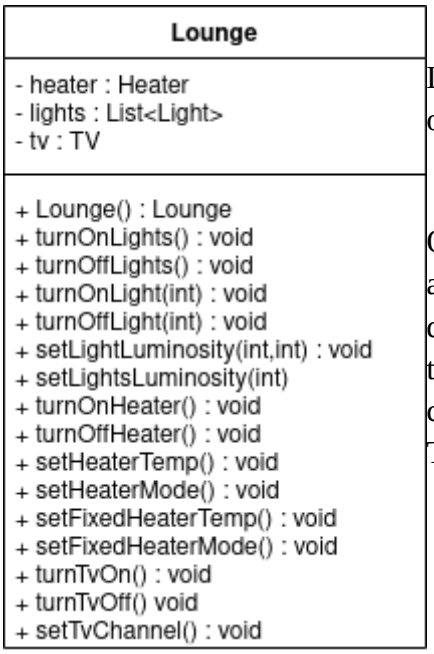


La classe presenta come attributi un oggetto AutomaticDoor e una Lista di Light.

Oltre al costruttore , la classe presenta i metodi per aprire/chiudere la porta automatica, accendere/spegnere una o tutte le luci, settare la luminosità di una o di tutte le luci e verificare lo stato di funzionamento dei dispositivi.

Salotto (Lounge)

Ecco il diagramma UML della classe Lounge:

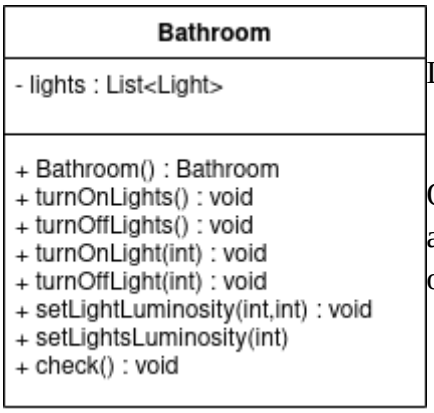


La classe presenta come attributi un oggetto di tipo Heater, un oggetto di tipo Tv ed una lista di Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di un o di tutte le luci, accendere/spegnere il condizionatore, settare la temperatura del condizionatore , settare la modalità del condizionatore, accendere/spegnere la Tv , settare il canale della Tv e verificare il funzionamento dei dispositivi.

Bagno (Bathroom)

Ecco il diagramma UML della classe Bathroom:

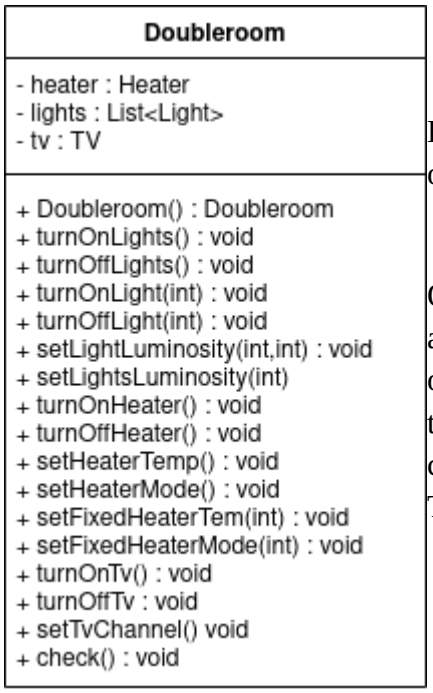


La classe presenta come attributi un lista di oggetti di tipo Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di una o di tutte le luci e verificare il funzionamento dei dispositivi.

Camera da letto matrimoniale(Doubleroom)

Ecco il diagramma UML della classe Doubleroom:

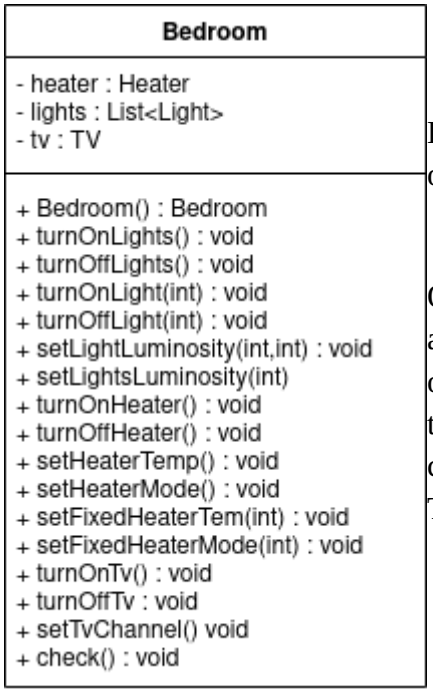


La classe presenta come attributi un oggetto di tipo Heater, un oggetto di tipo Tv ed una lista di Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di un o di tutte le luci, accendere/spegnere il condizionatore, settare la temperatura del condizionatore , settare la modalità del condizionatore, accendere/spegnere la Tv , settare il canale della Tv e verificare il funzionamento dei dispositivi.

Cameretta (Bedroom)

Ecco il diagramma UML della classe Bedroom:

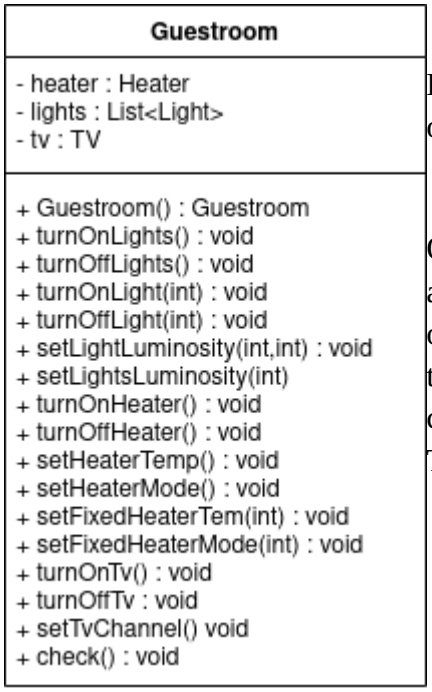


La classe presenta come attributi un oggetto di tipo Heater, un oggetto di tipo Tv ed una lista di Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di un o di tutte le luci, accendere/spegnere il condizionatore, settare la temperatura del condizionatore , settare la modalità del condizionatore, accendere/spegnere la Tv , settare il canale della Tv e verificare il funzionamento dei dispositivi.

Camera degli ospiti (Guestroom)

Ecco il diagramma UML della classe Guestroom:

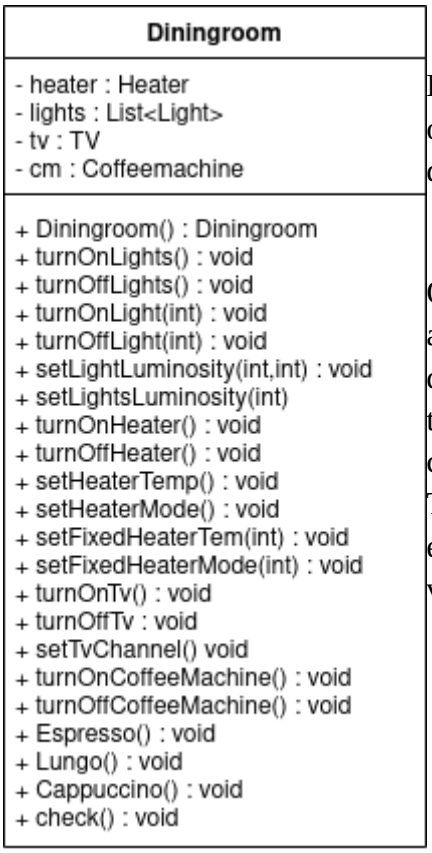


La classe presenta come attributi un oggetto di tipo Heater, un oggetto di tipo Tv ed una lista di Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di un o di tutte le luci, accendere/spegnere il condizionatore, settare la temperatura del condizionatore , settare la modalità del condizionatore, accendere/spegnere la Tv , settare il canale della Tv e verificare il funzionamento dei dispositivi.

Sala da pranzo (Diningroom)

Ecco il diagramma UML della classe Diningroom:

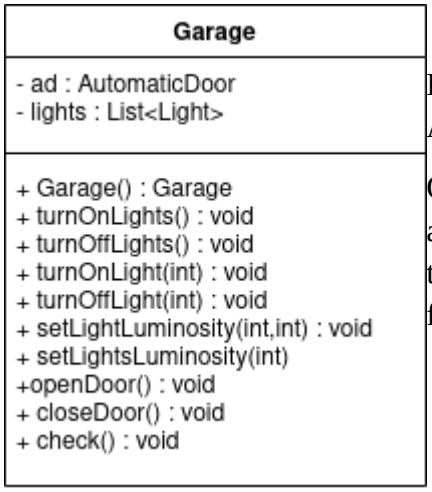


La classe possiede come attributi un oggetto di tipo Heater, un oggetto di tipo Coffeemachine, un oggetto di tipo TV ed una lista di oggetti di tipo Light

Oltre al costruttore , la classe presenta i metodi per : accendere/spegnere una o tutte le luci, settare la luminosità di un o di tutte le luci, accendere/spegnere il condizionatore, settare la temperatura del condizionatore , settare la modalità del condizionatore, accendere/spegnere la Tv , settare il canale della Tv, accendere/spegnere la macchinetta per il caffè, preparare un espresso, preparare un caffe lungo , preparare un cappuccino e verificare il funzionamento dei dispositivi.

Garage

Ecco il diagramma UML della classe Garage:

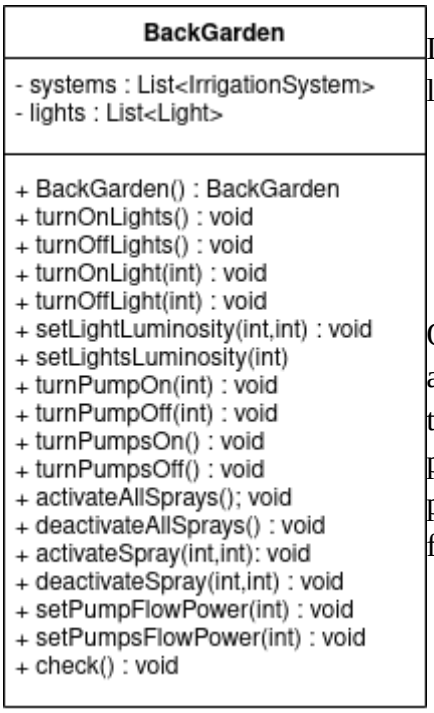


La classe presenta due attributi : un oggetto del tipo AutomaticDoor e una lista di oggetti Light

Oltre al costruttore , la classe presenta i metodi per: accendere/spegnere tutte o alcune luci, settare la luminosità di tutte o alcune luci, aprire/chiudere la porta e verificare il funzionamento dei dispositivi

Giardino posteriore (BackGarden)

Ecco il diagramma UML della classe BackGarden:

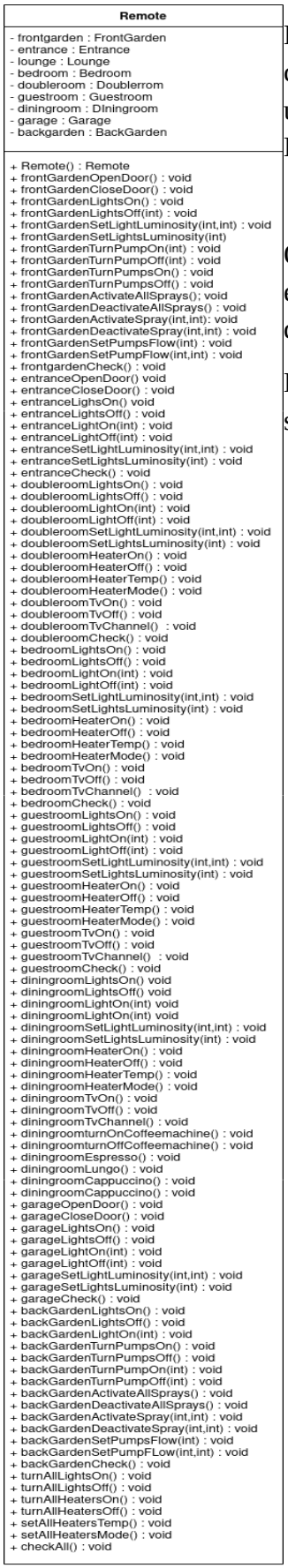


La classe presenta come attributi una lista di oggetti Light e una lista di oggetti IrrigationSystem

Oltre al costruttore, la classe presenta metodi per: accendere/spegnere tutte o alcune luci , settare la luminosità di tutte o di alcune luci, accendere/spegnere tutte o solo alcune pompe , attivare/disattivare tutti o alcuni spruzzi , settare la potenza di tutte o solo di alcune pompe e verificare il funzionamento dei dispositivi.

Telecomando (Remote)

Ecco il diagramma UML della classe Remote:



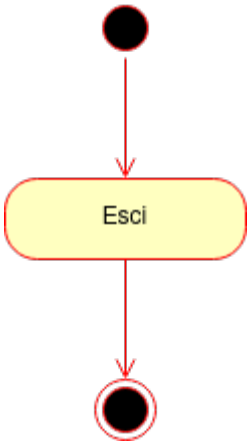
La classe Remote è la classe con cui il client si interfaccia. Essa possiede come attributi un oggetto FrontGarden, uno Entrance, uno DoubleRoom , uno Lounge, uno Guestroom, uno Garage , uno Bathroom , uno Bedroom, uno Diningroom e uno BackGarden.

Oltre al costruttore, la classe presenta svariati metodi per la gestione efficace dei dispositivi all'interno dei vari ambienti. Tutte le operazioni descritte in precedenza per le varie classi, sono qui riportate nuovamente.

La classe rappresenta l'interfaccia con cui dialogherà il Main, per cui al suo interno vi sono inserite tutte le operazioni svolgibili dal programma.

Diagrammi delle attività

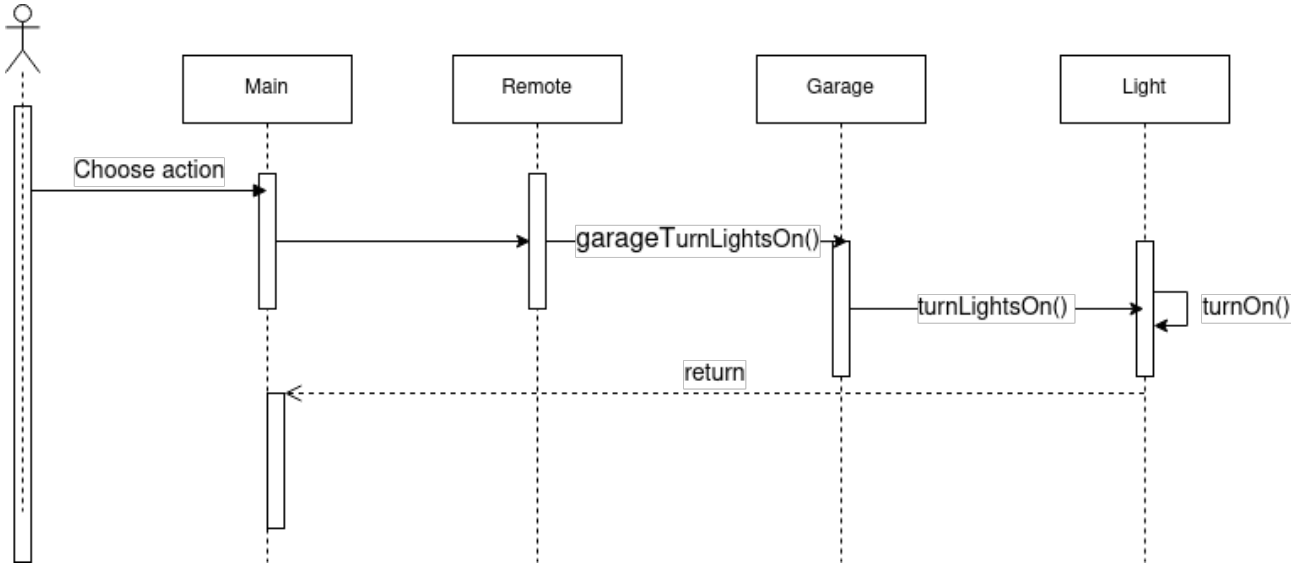
Come detto in precedenza , nella sezione di progettazione concettuale , analizziamo i diagrammi delle attività corrispondenti ai vari casi d’uso descritti. Poiché la quasi totalità delle operazioni si svolgono con la medesima sequenza di azioni, analizziamo solamente i casi generali.



Diagrammi di sequenza

Verranno mostrati solo due scenari poiché ,come detto in precedenza, vi è una possibile generalizzazione delle azioni. Cambierà solo la classe che lancerà il metodo.

Scenario 1 : accensione luci garage



Scenario 2 : uscita immediata