Università degli Studi di Napoli Parthenope

Dipartimento di Scienze e Tecnologie

Ingegneria del Software

Team 1

Smart Home

Docente:
Prof. A. Staiano

Studenti:
Martina Buffolino – 0124001850
Antonio Colucci – 0124001929
Jonathan De Michele - 0124001966
Antonio Sarno - 0124001914
Salvatore Starace – 0124001869

Anno accademico 2020 – 2021

Sommario

1 Introduzione

1.1 Scopo del Sistema

Il progetto Smart Home è stato commissionato con le specifiche di permettere la gestione di dispositivi intelligenti all'interno di un'abitazione.
Il sistema sarà intuitivo da usare quindi chiunque possieda dispositivi intelligenti può usufruire in maniera rapida ed efficiente del sistema.
Permetterà di creare delle stanze con una serie di dispositivi scelti dall'utente e di automatizzarli in modo da rendere molto più comoda e performante la vita di tutti i giorni.
L'utente avrà la possibilità di consultare con pochi semplici click tutti i dati relativi ai dispositivi configurati. Secondo le specifiche, il sistema dovrà mettere a disposizione dell'utente una serie di protocolli di sicurezza in caso di emergenza.

1.2 Obiettivi di progettazione

Il progetto Smart Home è stato sviluppato per semplificare le procedure di gestione di dispositivi intelligenti all'interno di un'abitazione, configurati direttamente dall'utente.
Grazie ad interfacce intuitive è possibile interagire e monitorare il sistema.
Il sistema Smart Home viene gestito dall'utente, pertanto è stato scelto di adottare un sistema di autenticazione che controlla l'accesso tramite l'inserimento di e-mail e password. Il progetto è stato realizzato nel linguaggio di programmazione Java, con paradigma di programmazione ad oggetti.

2 Architettura software attuale

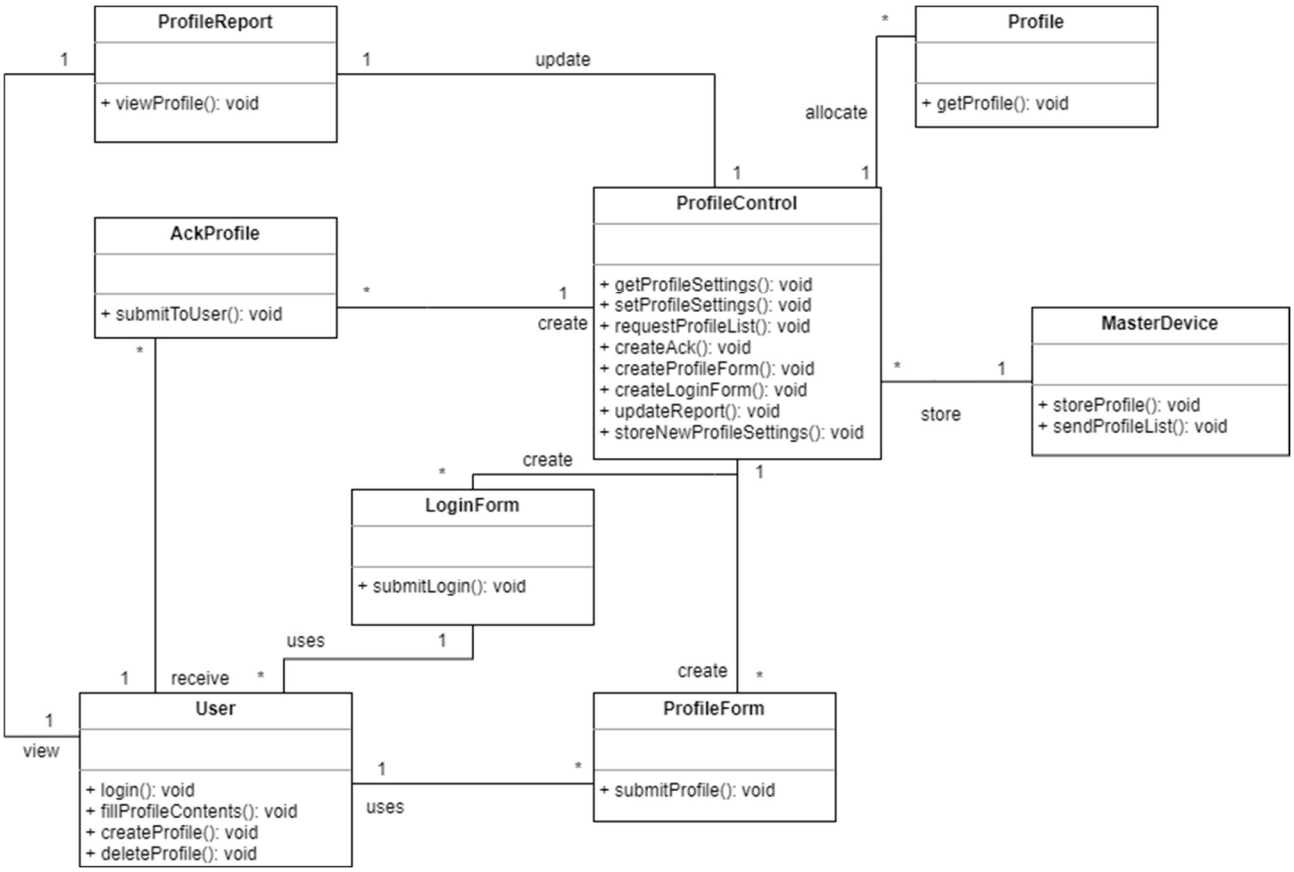
Nella realizzazione del sistema i progettisti non si sono potuti avvalere totalmente della possibilità di confronto con altri software simili in quanto i sistemi software presenti sul mercato, che permettono il monitoraggio dei dispositivi intelligenti, non consentono la gestione dello stato di esecuzione o manutenzione e non vi è la possibilità di gestire casi di emergenza.

3 Architettura software proposta

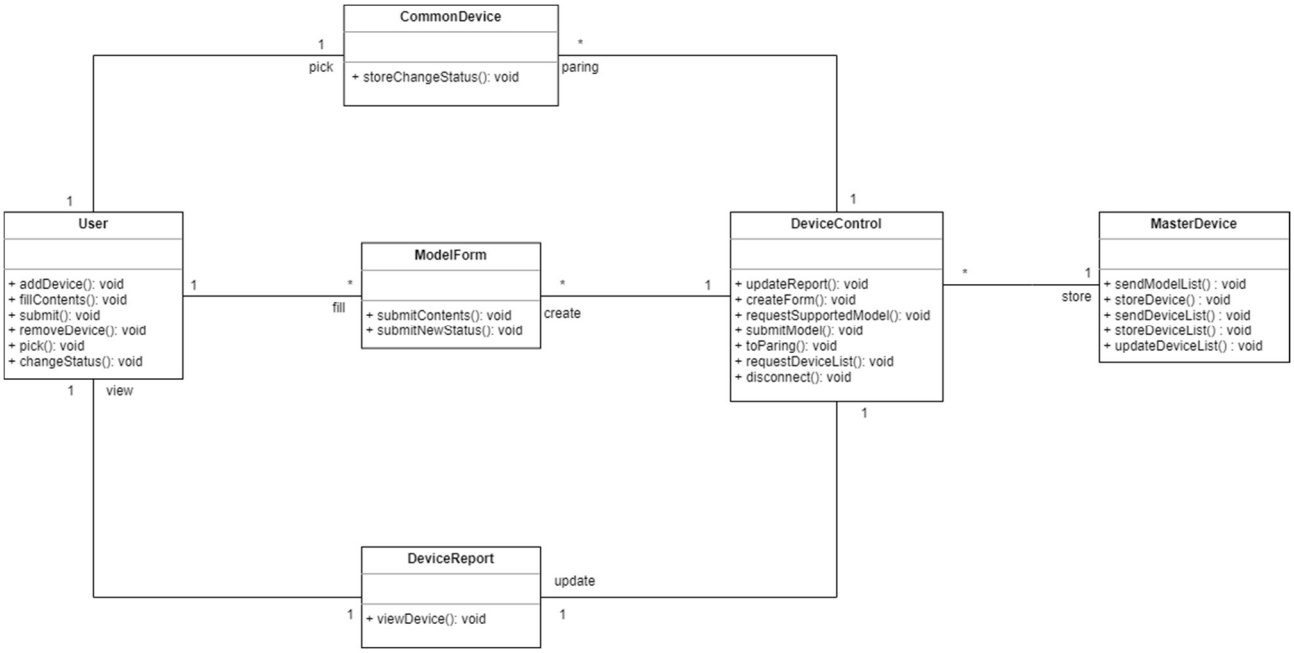
Per ovviare al problema dell'alto accoppiamento è stato scelto di adottare come stile architetturale il Model-View-Controller (MVC).
L'idea base dietro questo stile è di disaccoppiare l'accesso ai dati e la loro presentazione.
Al suo interno troviamo tre sottosistemi ovvero il model, che rappresenta lo stato del dominio applicativo, il view, che si occupa della visualizzazione delle informazioni all'utente, e il controller, che si occupa dell'interazione con l'utente e dell'aggiornamento delle view.
L'utilizzo del Model-View-Controller permette un riuso dei componenti, una miglior manutenzione e ottimizza il processo di testing ma allo stesso tempo si introduce un numero troppo elevato di classi per garantire la separazione.

3.1 Diagramma delle classi

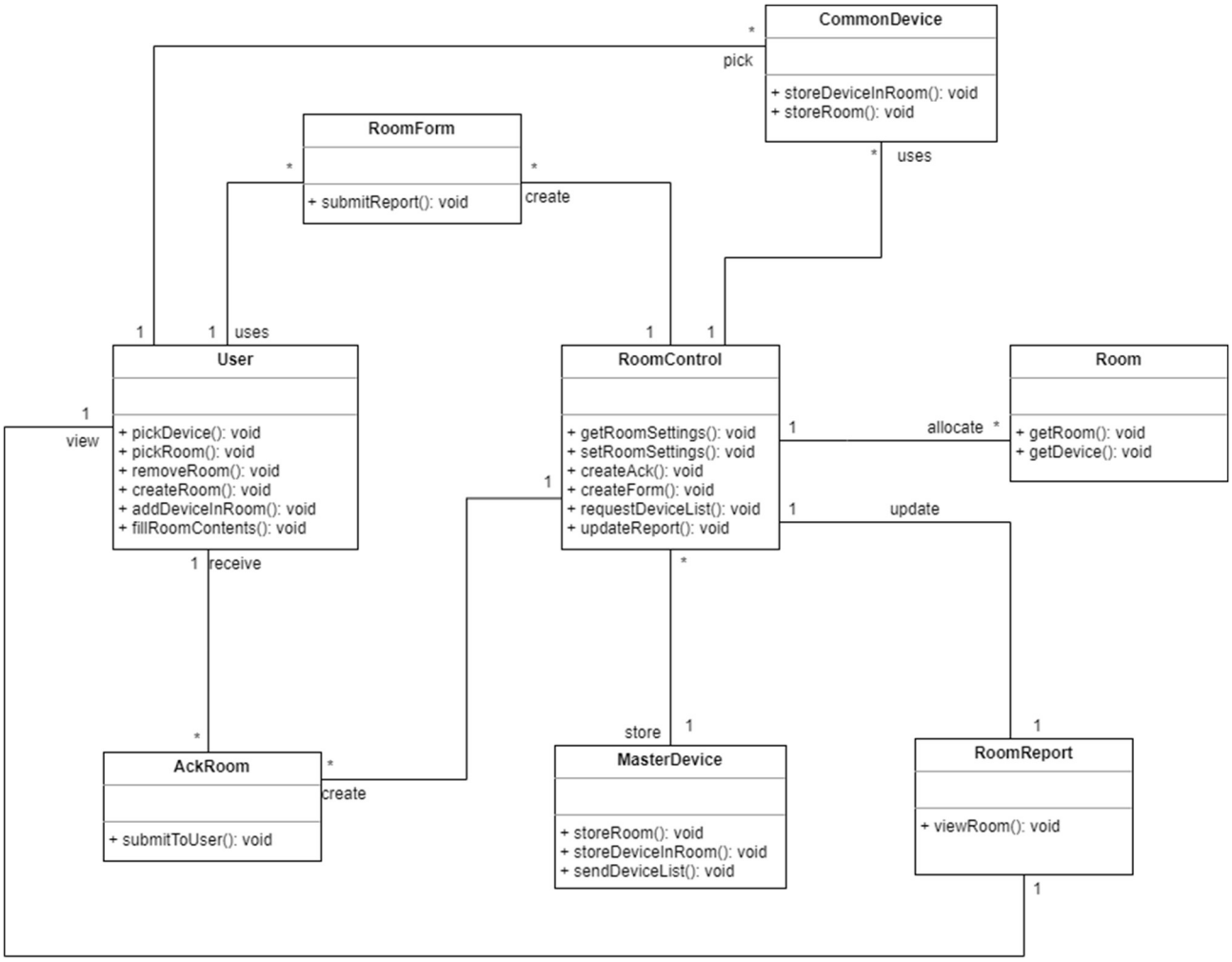
3.1.1 Classi Profile



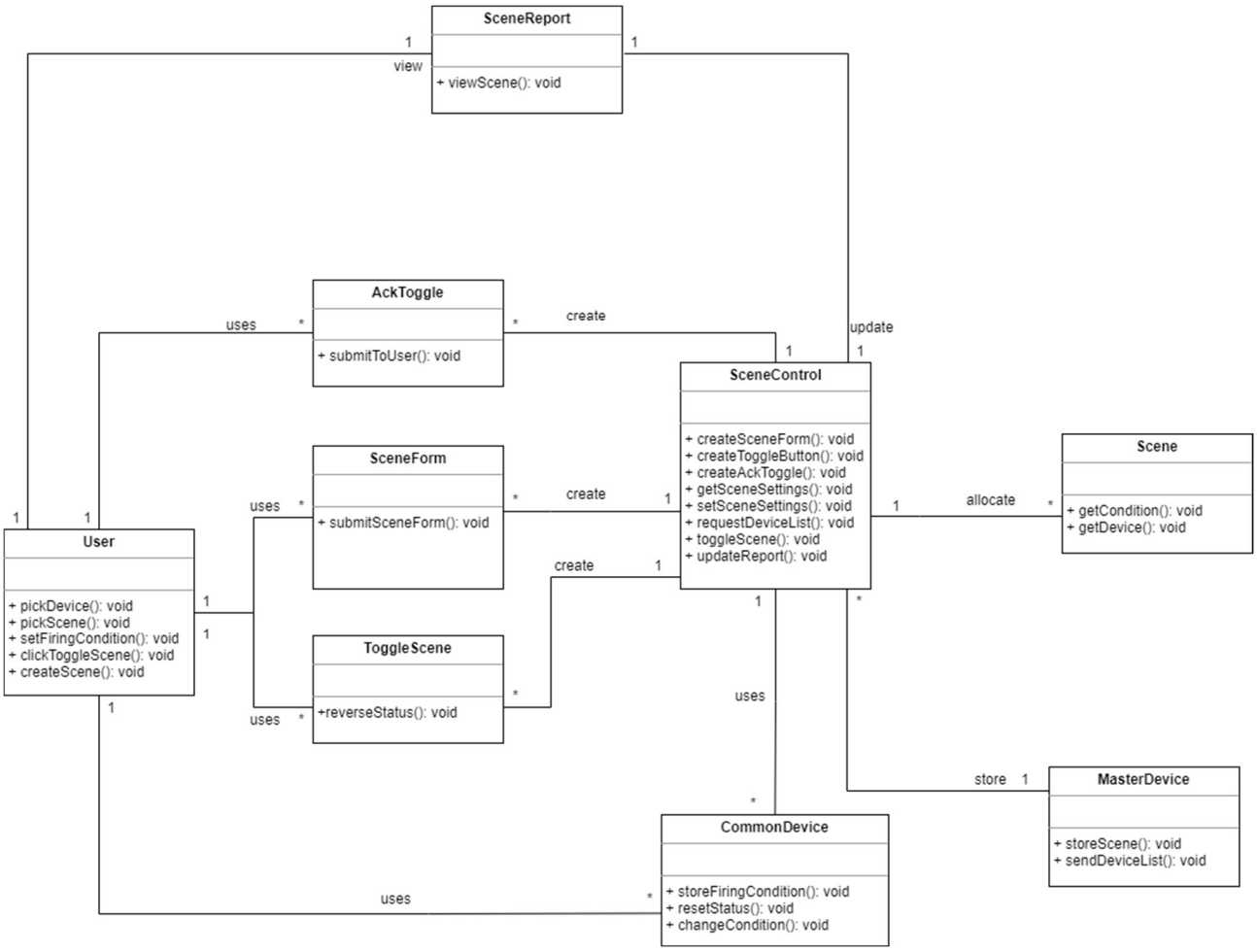
3.1.2 Classi Device



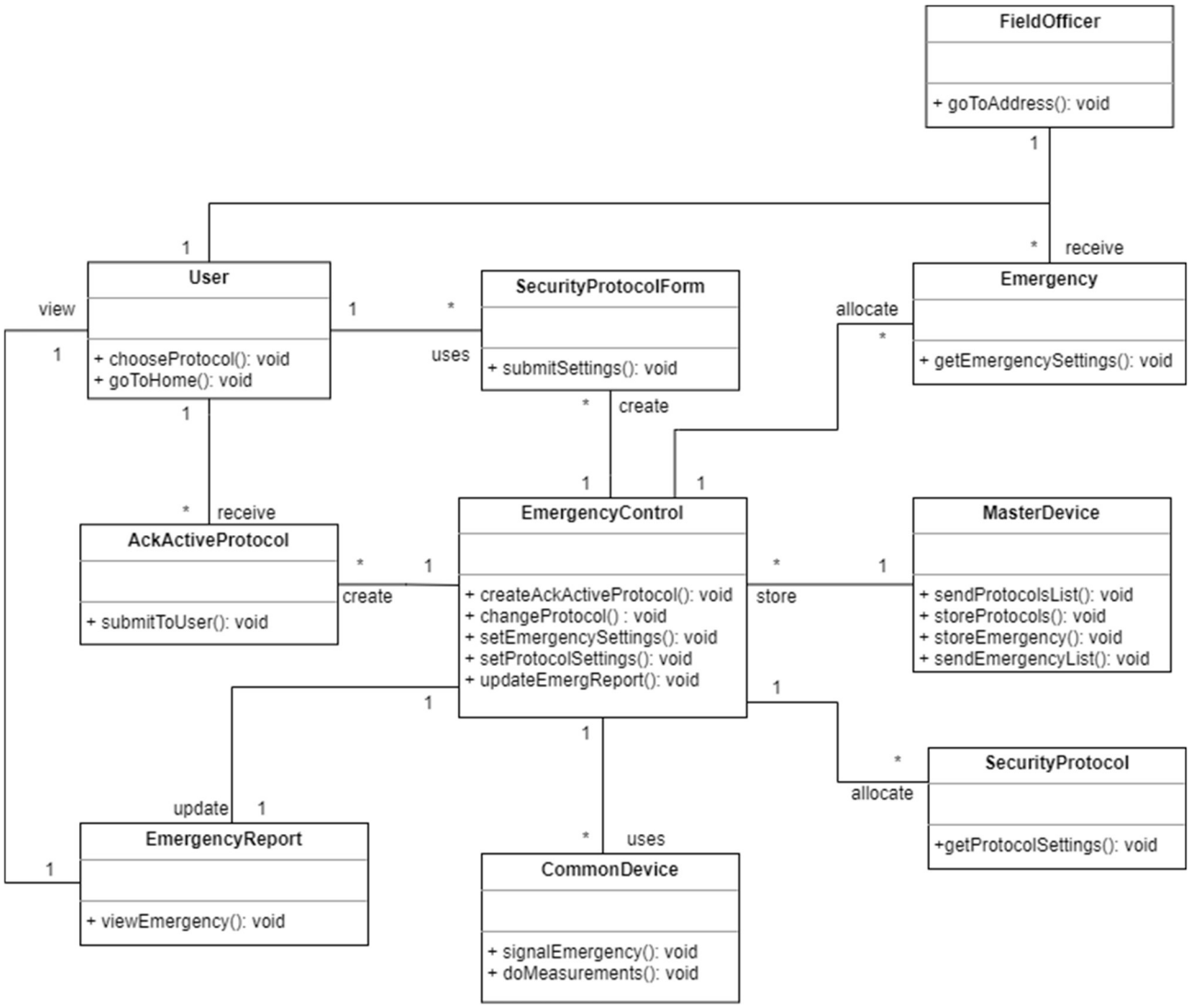
3.1.3 Classi Room



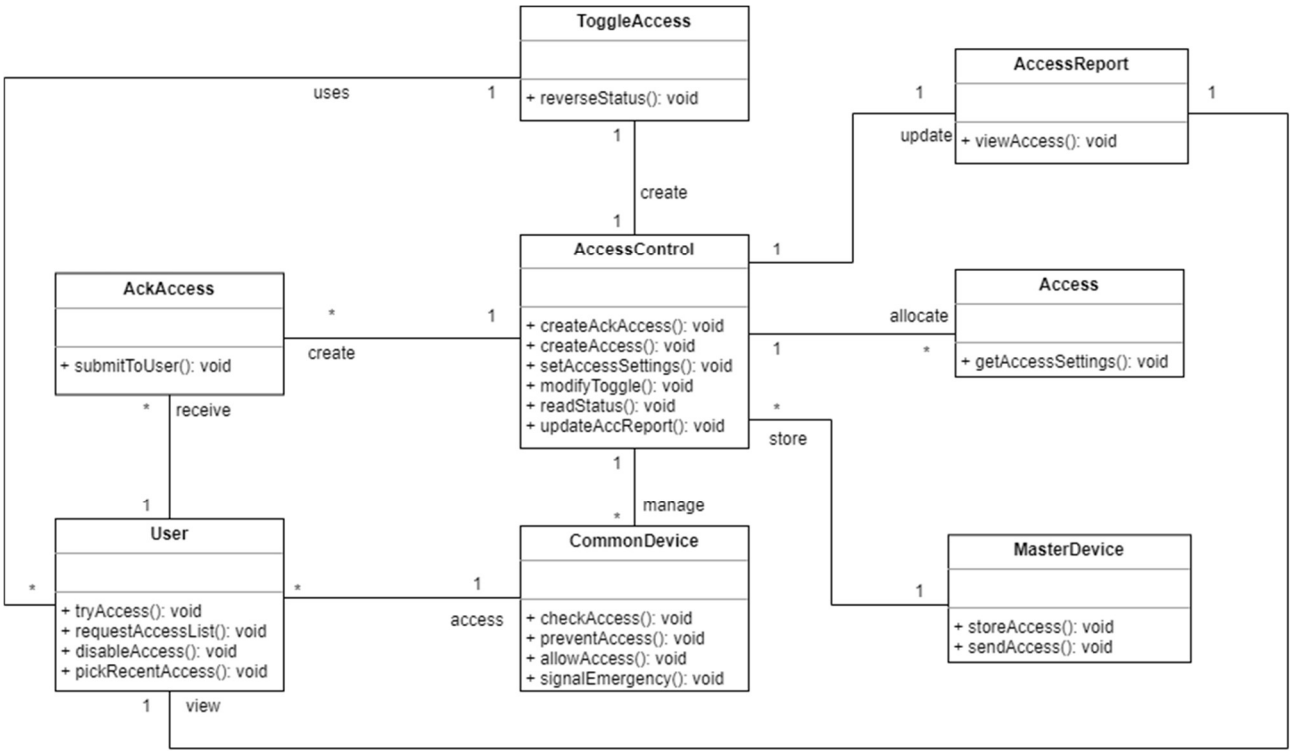
3.1.4 Classi Scene



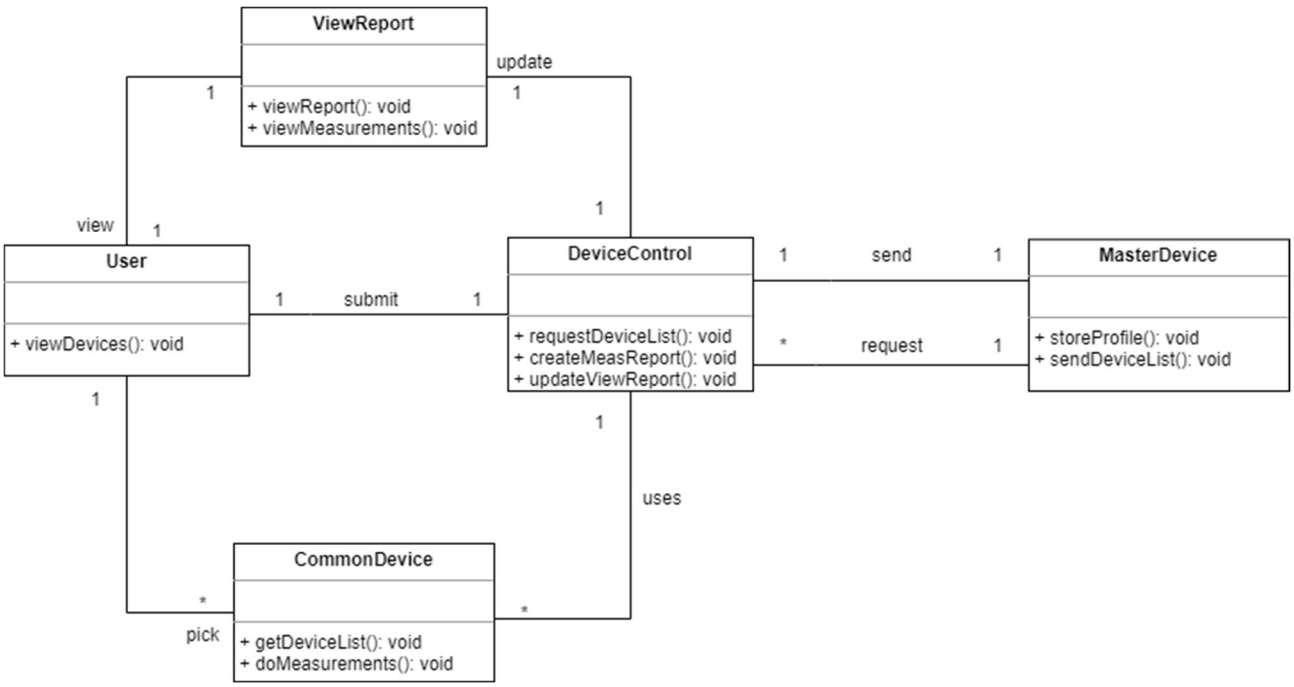
3.1.5 Classi Emergency



3.1.6 Classi Access



3.1.7 Classi View



3.2 Decomposizione in sottosistemi

All'interno dello stile architetturale Model-View-Controller sono stati individuati tre livelli di sottosistemi.
Il primo livello riguarda i sottosistemi presenti nel lato Model, il secondo livello quelli del lato View, e il terzo livello quelli del lato Controller.

Per quanto concerne il livello **Model** sono stati individuati al suo interno i seguenti sottosistemi che hanno il compito di gestire i dati del sistema:

1. Profile
2. Device
3. Scene
4. Room
5. Emergency
6. Access
7. CommonDevice
8. MasterDevice

Il lato Model, attraverso operazioni di aggiornamento, comunica con i sottosistemi del lato View, il cui compito è quello di rappresentare un qualsiasi output di informazione.

I sottosistemi individuati nel lato **View** sono:

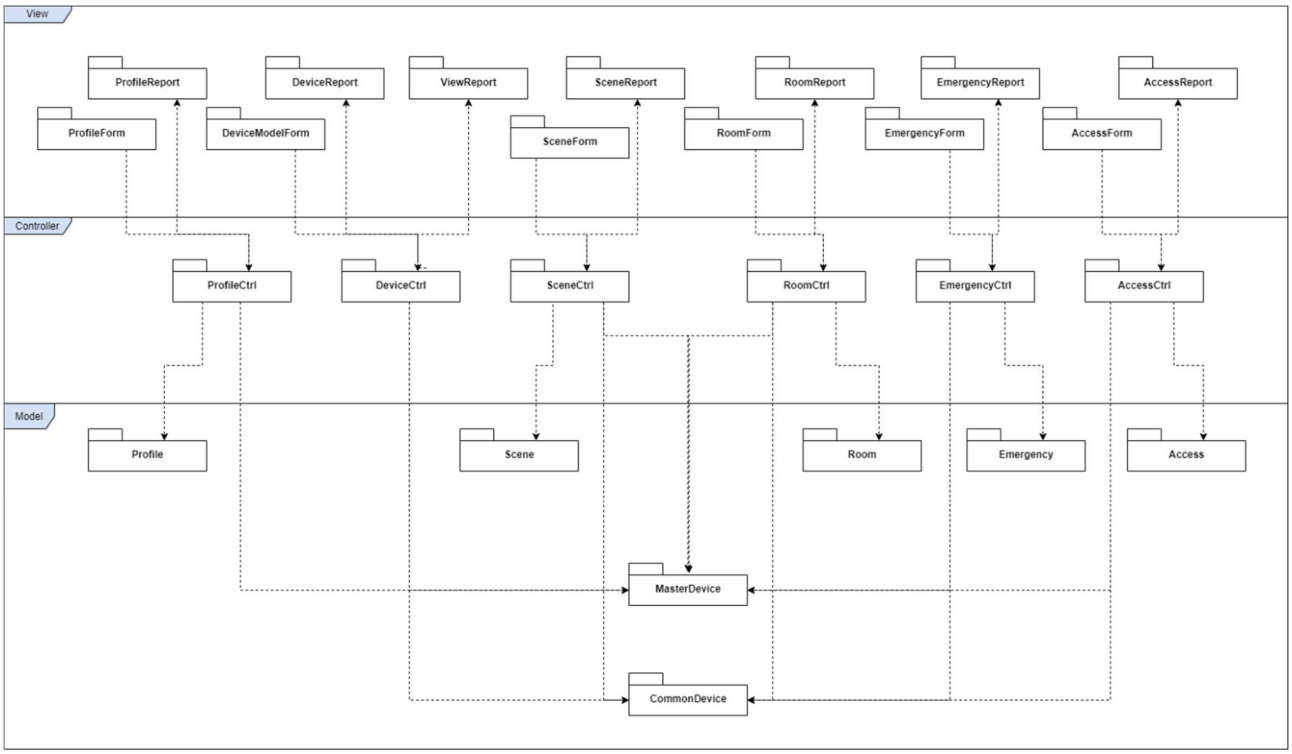
1. I "Report" che permettono la visualizzazione delle informazioni aggiornate all'utente (ProfileReport, DeviceReport, SceneReport, RoomReport, EmergencyReport, AccessReport, ViewReport).
2. I "Form", che vengono presentati all'utente attraverso moduli da compilare (ProfileForm, DeviceModelForm, SceneForm, RoomForm, EmergencyForm, AccessForm).

Il terzo livello comprende i sottosistemi del lato **Controller** che accettano l'input dall'utente e lo convertono in comandi per il Model e il View.

Essi sono:

1. ProfileCtrl
2. DeviceCtrl
3. SceneCtrl
4. RoomCtrl
5. EmergencyCtrl
6. AccessCtrl
7. ViewCtrl

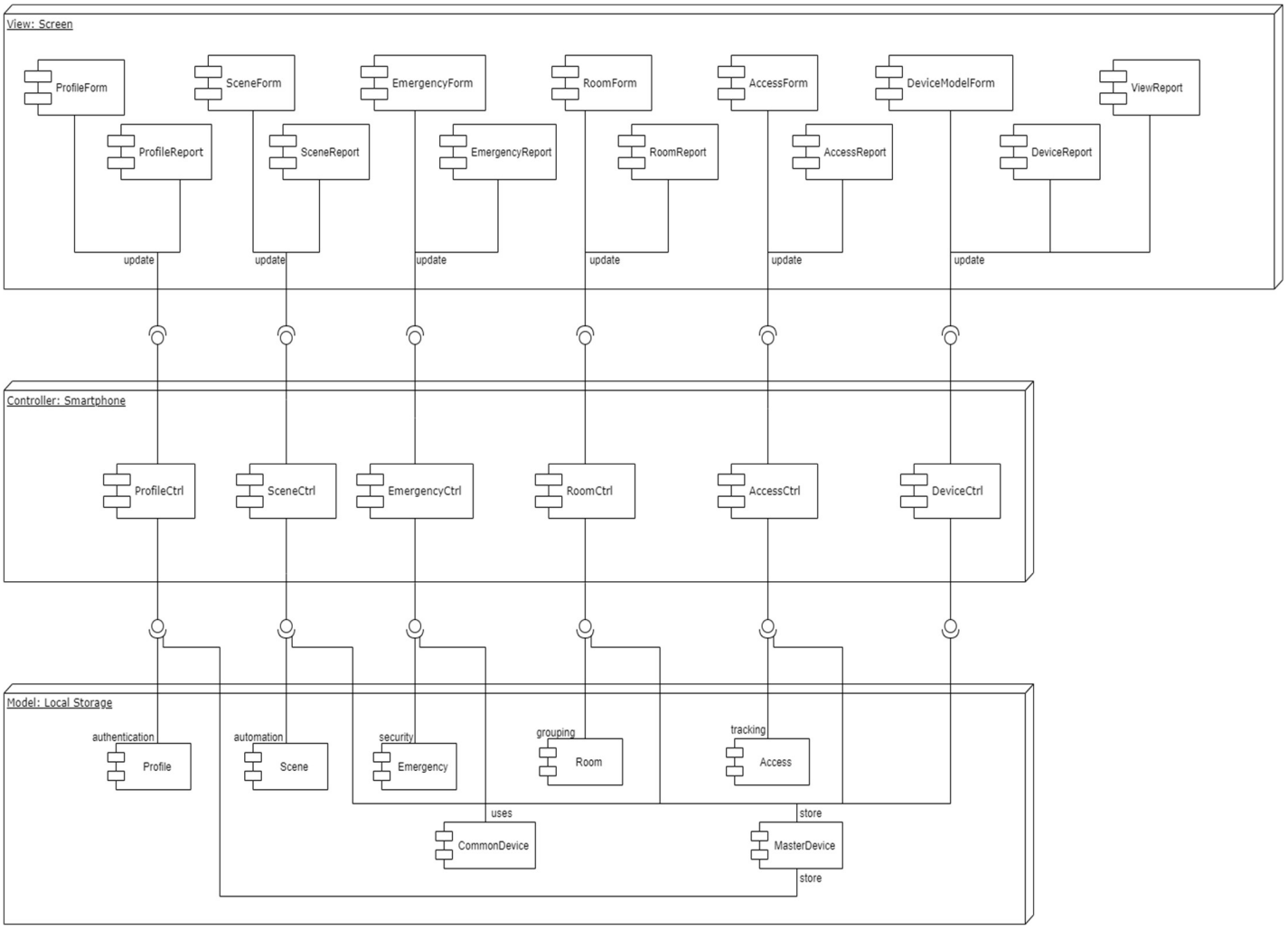
Il diagramma seguente illustra la suddivisione sopra descritta, mostrando le dipendenze tra i vari sottosistemi.



3.3 Mapping hardware/software

La struttura hardware “Smart Home” è costituita da tre componenti UML: un Model, un Controller e un View.
La componente Controller è costituita da oggetti Control che hanno il compito di fornire operazioni necessarie per le componenti View e Model. Quest’ultime dovranno effettuare richieste al Controller per eseguire le operazioni legate alle loro rispettive funzionalità.
La componente View ha al suo interno gli oggetti Report, attraverso i quali l’utente può visualizzare il sistema, e gli oggetti Form, che rappresentano i moduli da compilare che vengono presentati all’utente.
Nella componente Model, invece, troviamo gli oggetti che gestiscono i dati dell’intero sistema, tra cui il Master Device che si occupa principalmente di memorizzare le informazioni dell’intero sistema.

Di seguito è illustrato il diagramma di distribuzione con i componenti suddivisi per ogni nodo:



3.4 Gestione dei dati persistenti

Il sistema Smart Home si interfaccia con il MasterDevice, che è il perno centrale di tutte le operazioni, e al cui interno sono archiviati tutti i dati a disposizione per l'utente e per i dispositivi ad esso collegati. Il Master Device, inoltre, ha il compito di memorizzare i dati relativi ai dispositivi installati, le stanze che ne fanno parte, gli scenari di automazione creati dall'utente e il protocollo di sicurezza da seguire in caso di emergenza.

3.5 Controllo e sicurezza accessi

I permessi di accesso all'interno del sistema Smart Home sono garantiti dalle corrette credenziali di accesso, quali username e password che l'utente imposta all'atto della creazione del profilo.

Inoltre l'accesso all'interno dell'abitazione è consentito dall'inserimento di un codice PIN in un dispositivo.

Qualora il codice di accesso fosse inserito più volte in maniera sbagliata, il sistema azionerà il protocollo di sicurezza impostato, che notifica l'utente e il Field Officer del tentativo di autenticazione sospetto.

3.6 Gestione risorse globali

La progettazione dello Smart Home è decentralizzata in quanto il controllo è distribuito tra i vari oggetti.

3.7 Condizioni limite

In fase di inizializzazione devono essere acceduti i dati relativi al profilo dell'utente.

L'interfaccia utente all'avvio permette il login oppure la creazione di un nuovo profilo.

Si prevede che il sistema possa gestire 3 tipologie di guasti che verranno successivamente implementati:

1. Perdita di connessione, in cui i dispositivi installati entrano nella modalità stand-by.
2. Guasto di uno o più dispositivi comuni che presentano uno stato indefinito.
3. Guasto del MasterDevice, in seguito al quale il sistema offre prestazioni ridotte, ma che tuttavia consentono all'utente di effettuare le operazioni di base.