

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IASI

FACULTATEA DE INFORMATICA



LUCRARE DE LICENTA

**Home Smartify - soluție pentru controlul
dispozitivelor și securitatea casei**

propusă de

Liviu-Andrei Petrache

Sesiunea: iunie, 2023

Coordonator științific

Lect. dr. Vidrașcu Cristian

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IASI

FACULTATEA DE INFORMATICA

**Home Smartify - soluție pentru
controlul dispozitivelor și securitatea
casei**

Liviu-Andrei Petrache

Sesiunea: iunie, 2023

Coordonator științific

Lect. dr. Vidrașcu Cristian

Avizat,

Îndrumător lucrare de licență,

Lect. dr. Vidrașcu Cristian.

Data:

Semnătura:

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Petrache Liviu-Andrei** domiciliat în România, jud. Iași, mun. Iași, strada Stejar, numărul 41, bloc O10, scara A, apartament 3, parter, născut la data de 04 ianuarie 2001, identificat prin CNP 5010104226711, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2023, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Home Smartify - soluție pentru controlul dispozitivelor și securitatea casei** elaborată sub îndrumarea domnului **Lect. dr. Vidrașcu Cristian**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data:

Semnătura:

Declarație de consumămant

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Home Smar-tify - soluție pentru controlul dispozitivelor și securitatea casei**, codul sursă al pro-gramelor și celealte conținuturi (grafice, multimedia, date de test, etc.) care însășesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea "Alexandru-Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Liviu-Andrei Petrache**

Data:

Semnătura:

Cuprins

Motivătie	2
Introducere	4
1 Aplicații asemănătoare	5
1.1 Home Assistant	5
1.2 Apple Home	6
1.3 IFTTT	7
1.4 Concluzii	8
2 Tehnologii utilizate	9
2.1 Flutter	9
2.2 Flask	10
2.3 Raspberry Pi Zero W	10
2.4 Microcontrolere folosite	12
2.4.1 Arduino Nano	12
2.4.2 ESP8266	12
2.5 Module	14
2.6 Concluzii	14
3 Implementarea soluției	16
3.1 Pagina principală	17
3.1.1 Cărțile senzorilor	17
3.1.2 Locația senzorului	18
3.1.3 Modul de securitate	19
3.1.4 Butonul de refresh	20
3.2 Pagina senzorului static	21
3.3 Pagina senzorului dinamic	22

3.4 Pagina de Logs	23
3.5 Pagina de generare de rapoarte	24
3.6 Senzori statici	26
3.7 Senzori dinamici	28
Concluzii	29
Bibliografie	30

Motivație

Lucrarea de licență prezintă cum prin integrarea dispozitivelor "**Internet of things**"¹, se poate ajunge la un nivel ridicat de control asupra locuinței personale: monitorizarea temperaturii și umidității unei camere, conectarea/deconectarea unui consumator de la alimentare, sarcini multiple lansate în același timp, toate fiind ușor de executat cu ajutorul aplicației mobile.

Home Smartup a fost adusă la viață datorită dorinței personale de a-mi monitoriza și controla apartamentul într-un mod ușor, intuitiv și privat. Jucându-mă cu plăcile **Raspberry Pi**² împreună cu diferiți senzori, am reușit să creez un sistem ce securiza ușa camerei mele cu ajutorul unui cod numeric, roboți ce urmează un traseu predefinit și prize inteligente. Problema pe care am întâmpinat-o a fost că nu exista o entitate ce să comande fiecare modul în parte, eu fiind responsabil de a trimite informațiile necesare către modulul respectiv. Observând utilitatea acestei funcționalități, faptul că mă distrez în timp ce programez și conectez componentele între ele prin fire, am decis să realizez un sistem ce poate "*vorbi*" cu aceste module, putând fi comandate din aplicația de telefon. Inițial, ideea a fost concepută doar cu latura de monitorizare, folosind un singur tip de senzori. Oferit pe post de feedback de către domnul profesor coordonator Cristian Vidrașcu, am realizat că implementarea unei modalități de control a acestor dispozitive adaugă o nouă arie de funcționalitate și creativitate în gestionarea sarcinilor din casă.

¹<https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>.

²<https://www.raspberrypi.org/>.

Contribuții

Soluțiile de a face o casă “smart” cresc zilnic în număr, astfel am optat pentru a crea una originală, de la framework-uri, până la modulele hardware pe care le-am folosit. Menționez astfel că arhitectura software este creată de mine, librăriile folosite sunt open source, împreună cu design-ul modulelor hardware.

Aș dori să mulțumesc domnului profesor coordonator Cristian Vidrașcu pentru libertatea alegerii temei licenței și profesionalismul de care a dat dovadă.

Introducere

Proiectul este împărțit în X capituloare care detaliază fiecare element ce ajută la crearea aplicației:

Capitolul I. Aplicații asemănătoare - prezentarea unor soluții de pe piață ce rezolvă problema tratată în teză.

Capitolul II. Tehnologii utilizate

Capitolul III. Arhitectura + implementari specifice

Capitolul IV. Use case-uri

Capitolul V. Îmbunatatiri pe viitor

Capitolul 1

Aplicații asemănătoare

Acest capitol se focusează asupra prezentării unor soluții asemănătoare ca idee sau ca și tehnologie cu proiectul **Home Smartify**.

În prezent, există numeroase astfel de platforme, fiind alese cele mai preferate în rândul utilizatorilor care doresc să își adapteze casa după preferințele personale într-o manieră *Do it yourself (DIY)*¹

1.1 Home Assistant

Home Assistant (Figura 1.1) este o platformă open-source² pentru controlul și automatizarea locuințelor smart. Oferind un mediu centralizat pentru gestionarea dispozitivelor și serviciilor, ea suportă peste 1000 de servicii și device-uri diferite pe care le identifică prin scanarea initială a rețelei de internet la care sunt conectate.

Orice utilizator are asignat o *amprentă digitală*³ care este folosită spre a îl identifica în mediul online. Confidentialitatea constituie unul din punctele forte ale aplicației Home Assistant, astfel că păstrează toate informațiile în domeniul local și evită pe cât posibil comunicarea cu alte servicii din cloud.

Este recomandat ca acesta să fie instalat pe o platformă Raspberry Pi prin folosirea sistemului de operare pus la dispoziție pe website sau poate fi executat într-un *container Docker*⁴, metodă similară cu cea implementată în această licență.

¹<https://www.doityourself.com/>.

²<https://opensource.com/resources/what-open-source>.

³<https://www.kaspersky.com/resource-center/definitions/what-is-a-digital-footprint>.

⁴<https://www.docker.com/>.



Figura 1.1: Interfața meniului principal, temperatură, grafice, consum energie. Credits: <https://rbx.dk/w/home-assistant/hvad-er-home-assistant.html>

1.2 Apple Home

Apple este una dintre primele companii care au încercat să integreze controlul casei smart într-o aplicație mobile, lansând HomeKit (Figura 1.2) integrat în IOS 8 care a apărut în septembrie 2014.

Cu suport de peste 100 brand-uri de produse, printre care termostate, prize inteligente, lumini și jaluzele, utilizatorii de Mac, Iphone, Ipad își pot automatiza sarcinile, numite *Scenes*(care pot fi separate pe zone), prin apăsarea unui buton din interfața ușor de înțeles sau pot opta pentru integrarea cu asistentul vocal Siri.

Cristopher Null⁵ de la TechHive afirmă că procesul de instalare este impresionant, dar folosirea acestui program zilnic poate fi *mediocră*, fiind clasificată ultima când vine vorba de control din cauza pierderii semnalului și eşuarea executării task-urilor. Speră ca aplicația să primească update-uri și noi funcționalități, interacțiunea sa cu Apple Home lăsând mult de dorit. Se poate observa numărul de 10 ori mai mic de dispozitive suportate față de Home Assistant, fiind o barieră destul de observabilă atunci când se pune problema extinderii către alte funcționalități.

⁵<https://www.techhive.com/article/579157/essential-homekit-guide.html>.

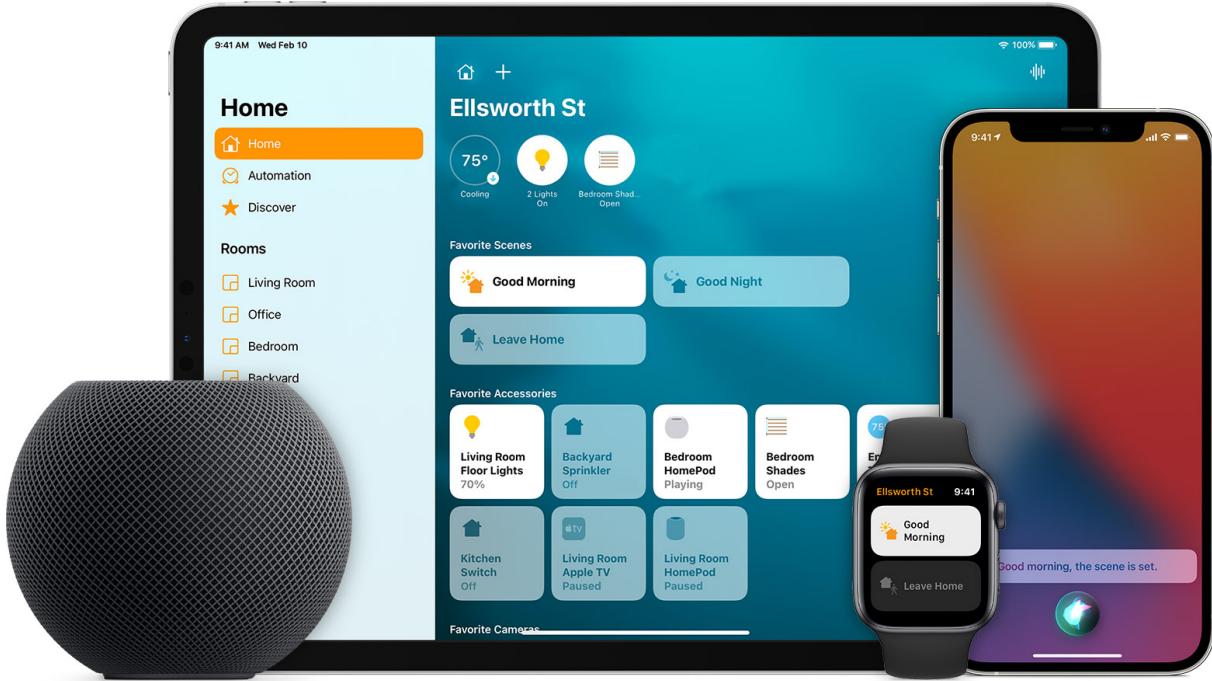


Figura 1.2: Meniul principal împreună cu dispozitivele ce sunt în control. Credits: <https://support.apple.com/en-us/HT208280>

1.3 IFTTT

Fie că dorești să fii notificat pe telefon cu o zi înaintea unui eveniment sau că vrei ca asistentul Google să îți schimbe intensitatea luminilor din garaj, acestea sunt exemplele de bază pe care platforma *IF this then that* (prescurtat **IFTTT**) (Figura 1.3) le poate realiza.

Acesta este un serviciu ce îți oferă posibilitatea de a conecta aplicații, servicii și dispozitive fără a fi preocupat de programarea lor. Website-ul IFTT și aplicația de telefon te ajută să construiești comenzi (pe care IFTTT obișnuia să le numească *rețete*, acum sunt *applets*) folosind card-uri colorate puternic pentru identificare.

Odată intrat pe pagina de start, se pot vedea applet-urile care sunt conectate cu ajutorul informațiilor de la capătul cardului. Pentru crearea unui astfel de item, trebuie specificat serviciul ce declanșează automatizarea respectivă. Odată ce se întâmplă acel trigger (*if this*), se vor executa acțiunile definite de către utilizator (*then that*). Ex: dacă ai adăugat o melodie nouă într-un playlist pe Spotify, adaug-o și pe Apple Music.

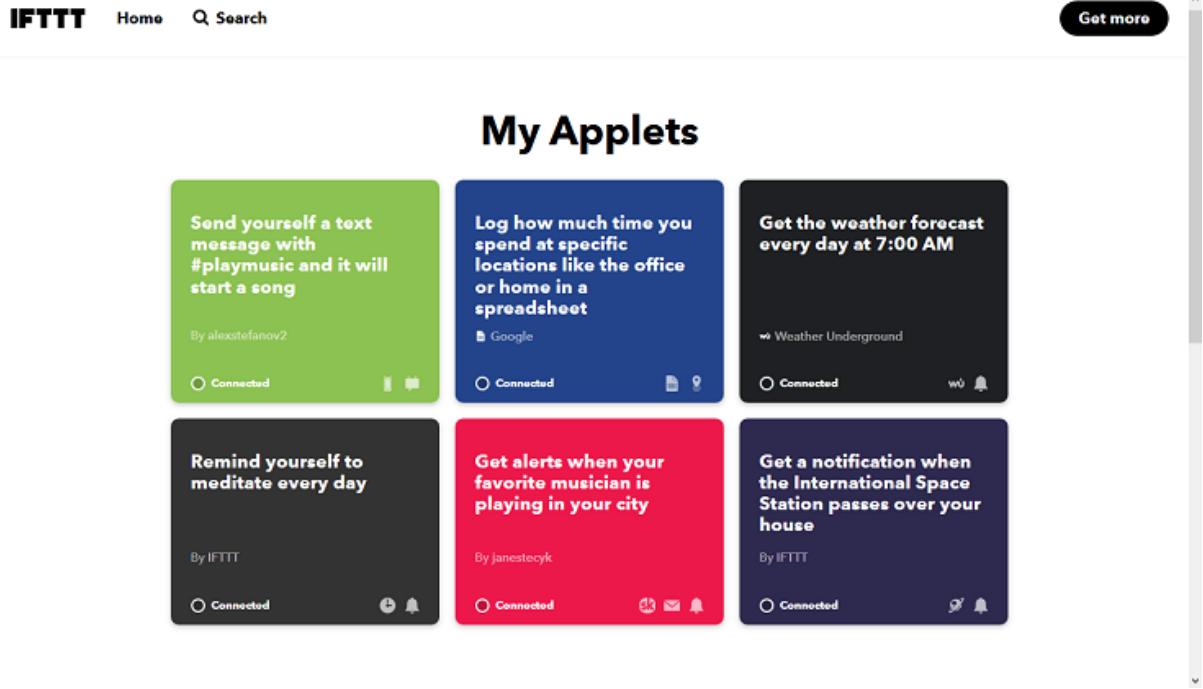


Figura 1.3: Cardurile colorate pentru diferențiere, cu descriere, status și autorul informației. Credits: Ben Moore @ <https://www.pcmag.com/reviews/ifttt>

1.4 Concluzii

După cum se poate observa din exemplele oferite mai sus, soluțiile pentru o casă smart sunt din ce în ce mai populare deoarece este foarte probabil ca un utilizator să dețină dispozitive IoT pe care le poate inter-conecta să obțină o utilizare eficientizată a spațiului de lucru.

Soluțiile pentru o casă smart oferă oportunități diverse de a îmbunătăți controlul și automatizarea locuinței. Fiecare platformă are propriile caracteristici și avantaje, iar utilizatorii pot alege în funcție de nevoile și preferințele lor specifice.

Capitolul 2

Tehnologii utilizate

2.1 Flutter

Flutter este o platformă pentru dezvoltare de aplicații publicată inițial în 2017. Cu ajutorul acestui **SDK** (*Software Development Kit*), programatorii pot construi aplicații web, desktop și cross-platform pentru Android și IOS. Se folosește împreună cu limbajul de programare Dart.

Pentru a dobândi o eficiență asemănătoare celei native, este folosită compilarea **AOT** (*ahead-of-time*) pe toate platformele în afară de Web, unde codul este convertit în JavaScript.

Fundația acestui SDK este scrisă în **C++**, oferă randare low-level folosind biblioteca *Skia* de la Google sau un strat de abstractizare grafic *Impeller*. Componenta de bază în Flutter este *widget-ul* care poate conține și alte widget-uri. Aceasta descrie logica, interacțiunea și design-ul unei componente UI. Flutter conține două pachete de elemente UI: Material Design (Google) și Cupertino (Apple).

```
1 return Scaffold(  
2   body: Column(  
3     children: [  
4       Text('Acest text va fi pozitionat'),  
5       Text('Deasupra acestui text'),  
6       ],  
7     ),  
8   );
```

Listing 2.1: Widget Scaffold

2.2 Flask

Flask este un framework web ușor de utilizat și flexibil, scris în Python. A fost creat cu scopul de a fi simplu și minimalist, oferind totodată funcționalitățile esențiale necesare pentru dezvoltarea rapidă a aplicațiilor. Aceasta permite definirea de rute pentru a mapa cererile HTTP/HTTPS primite către funcții. Prin intermediul adnotărilor, se pot atribui funcțiilor URL-uri și metode HTTP/S specifice, permitând astfel aplicației să proceseze cererea în funcție de parametrii primiți.

Datorită abordării sale minimalistă și comunității active, Flask a devenit unul dintre cele mai populare framework-uri pentru dezvoltarea aplicațiilor web în Python.

Figura 2.2 reprezintă un script ce expune o rută care returnează un paragraf HTML cu textul *Hello, World!*.

```
1 from flask import Flask  
2 app = Flask(name)  
3 @app.route("/")  
4 def hello_world():  
5     return "<p>Hello, World!</p>"
```

Listing 2.2: Exemplu minimal de aplicație Flask

2.3 Raspberry Pi Zero W

Raspberry Pi este o familie de calculatoare, de mărimea unui card de credit sau mai mici, care au revoluționat industria cu prețul accesibil de doar \$25 al primei plăci. În ultimii 5 ani, acestea au dobândit o atenție foarte mare în rândul comunității *Do it yourself (DIY)* datorită spectrului larg de utilizări în proiecte digitale și IoT.

Comunitatea Raspberry a crescut mult și consistent. În momentul de față, în urma căutării "Raspberry pi projects", ne sunt returnate 47.2 milioane de pagini care conțin tutoriale despre imprimante 3D, sisteme de irigare, de luat vederi, ad-blocker pentru rețea și multe altele.

În contextul versiunii Zero W (Figura 2.1), avem disponibilă o placă de mărimea unui pachet de gume: 6.5 x 3 x 0.5 cm pe care poate fi instalat un sistem de operare bazat pe GNU/Linux numit Raspbian OS. Are un procesor single-core de 1GHz, 512 MB LPDDR2 RAM, ieșire mini HDMI, două porturi micro USB, Bluetooth 4.0 și 2.4GHz 802.11n Wi-Fi.

Aspectul special al acestor computere constă în cei 40 pini **GPIO** (*General purpose input-output*). Aceștia sunt folosiți pentru conectarea de senzori, fiind ușor programabili cu ajutorul limbajului Python, sau pentru audiențe tinere, Scratch.

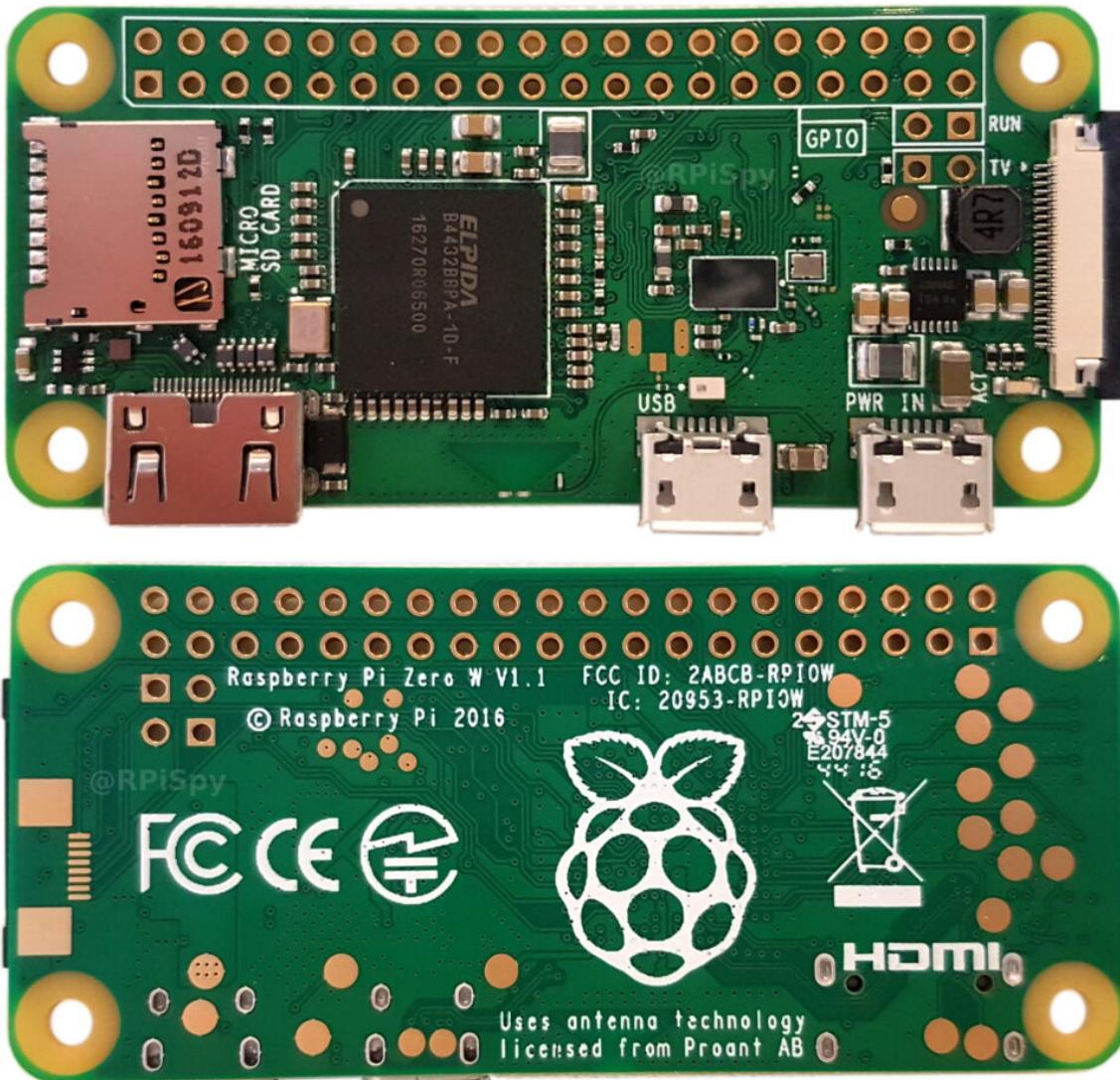


Figura 2.1: Față și spatele plăcii Raspberry Pi Zero W Credits: RPISpy @ <https://www.raspberrypi-spy.co.uk/>

2.4 Microcontrolere folosite

Un microcontroller (abreviat *MCU*) este un circuit integrat care combină funcționalitate microporcesorului (de obicei aflat sub un radiator de aluminiu pentru disiparea căldurii) cu componentele periferice: module *Input/output*, memorie și interfețe de comunicare cu alte module. Dețin mai mulți pini de tip analog, digital și **PWM** (*Pulse Width Modulation*) pentru conectarea cu alte module

2.4.1 Arduino Nano

Arduino Nano (Figura 2.2) este fratele mai mic al plăcii Uno cu care împărtășește majoritatea funcționalității: microcontroller ATmega328, 32KB memorie flash, viteză de 16MHz și 22 pini I/O. Singurele diferențe sunt mărimea redusă și portul de USB mini. Este perfect pentru persoanele care doresc să învețe tainele electronicii și a programării, fiind potrivit pentru proiecte ce au constrângeri legate de spațiul disponibil.

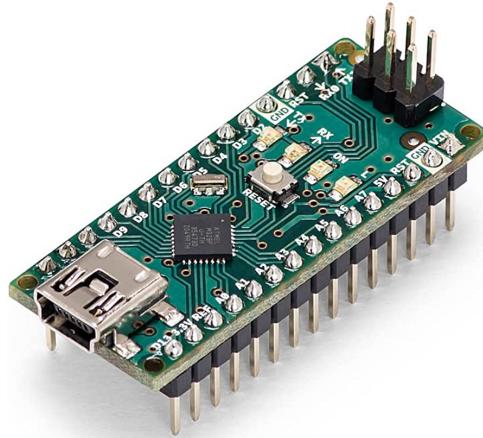


Figura 2.2: Arduino Nano

<https://www.amazon.com/Arduino-A000005-ARDUINO-Nano/dp/B0097AU5OU>

2.4.2 ESP8266

Modulul ESP8266 (Figura 2.3) este un **SOC** (*System on a chip*) produs de firma Espressif Systems ce folosește stack-ul TCP/IP cu ajutorul căruia se poate conecta la internet.

Acesta a fost popularizat în anul 2014 de către alt SOC asemănător **ESP-01** al firmei **Ai-Thinker**. Chip-urile se pot conecta la rețele Wifi și să realizeze conexiuni

prin utilizarea comenzilor de tipul *Hayes, AT commands*.

La început, nu existau documentații în limba engleză. Dat fiind faptul că SoC- ul este ieftin și deține un număr mic de componente externe, acesta poate fi produs în volume mari la preț mic, ceea ce a atras mulți hackeri să-l exploreze și să traducă documentația.

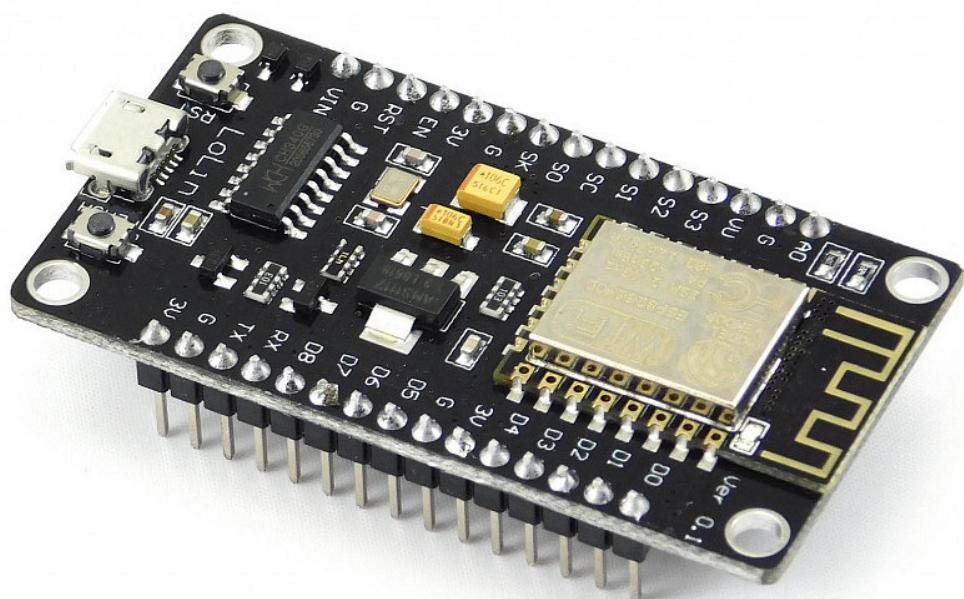


Figura 2.3: Microcontroller-ul ESP8266 (chip-ul argintiu) pe o placă de tipul *Breakout*
<https://www.teachmemicro.com/esp8266-spiffs-web-server-nodemcu/>

2.5 Module

Modulul este echipat cu 2 relee (Figura 2.4a) controlate individual. Acest modul poate fi utilizat împreună cu orice placă de dezvoltare care dispune de 2 pini digitali și un pin VCC de 5 V. Produsul este util în multe proiecte de electronică în care trebuie controlate diferite dispozitive care se alimentează cu o tensiune maximă de 250 V AC sau 30 V DC.

Senzorul DHT22 (Figura 2.4b) este un senzor de umiditate și temperatură mic, ieftin și la îndemână. Folosește un senzor capacativ de umiditate și un termistor pentru a măsura aerul din jur. Acesta oferă semnal digital pe pinul de date. Este ușor de folosit, dar este nevoie de atenție pentru citirea datelor la momentul potrivit.

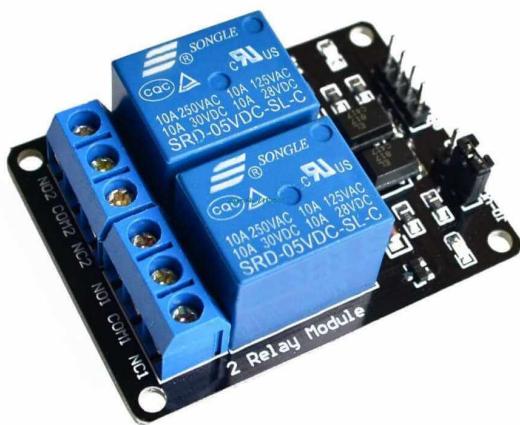
Senzorul infraroșu HC-SR501 (Figura 2.4c) este folosit pentru a detecta prezența oamenilor. Este des utilizat în aprinderea sau stingerea automată a luminii într-o încăpere atunci când o persoană ajunge sau părăsește o incintă. Suportă o tensiune de alimentare mai ridicată (5 – 20 V) și are un consum mai mare. Are o rază de sensibilitate de 110° și detectează obiecte până la o distanță de 7m.

Placa HC-SR04 (Figura 2.4d) deține un emițător și un receptor ultrasonic ce ne pot oferi distanța până la cel mai apropiat obstacol într-un mod ușor de programat. E compatibilă cu o gamă variată de plăci de dezvoltare, având unele avantaje față de senzorii de distanță de tip analog: are nevoie doar de pini I/O digitali, are toleranță mai mare la zgomotul din circuit și precizie de până la 4.5 metri, toate la un consum continuu de maxim 2mA.

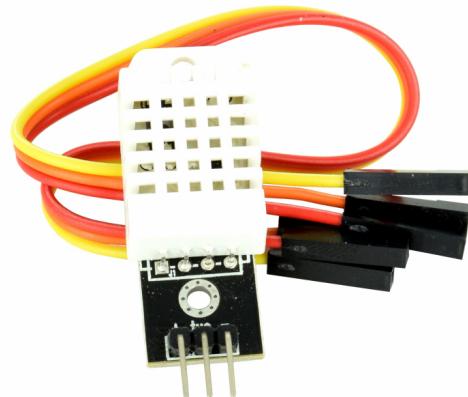
2.6 Concluzii

Domeniul Smart Home este foarte vast și oferă o gamă largă de oportunități creative. Tehnologia a evoluat rapid, permitând controlul și automatizarea diverselor aspecte ale locuinței, de la iluminat și securitate la gestionarea energiei și confortul locatarilor.

Modulurile și plăcile folosite se pliază destul de bine pentru aplicările descrise în viitorul capitol, dar întotdeauna acestea vor fi actualizate cu noi funcționalități. Prin urmare, în domeniul caselor inteligente, nu există limite, iar utilizatorii sunt încurajați să-și folosească imaginația și să exploreze noi soluții ce se pliază vieții personale.



(a) Placă cu două relee pentru comutarea tensiunii prizei



(b) Senzor de temperatură și umiditate



(c) Senzor de detectarea mișcării



(d) Senzor de măsurare a distanței

Figura 2.4: Senzorii folosiți în acest proiect

Capitolul 3

Implementarea soluției

Prin parcurgerea acestui capitol, cititorul va avea o imagine de ansamblu asupra modului în care senzorii sunt integrați în aplicație, cum este realizată comunicarea cu serverul și care sunt aspectele legate de programarea și funcționalitatea aplicației mobile. Aceste informații sunt esențiale pentru a înțelege arhitectura (Figura 3.1) și funcționalitatea aplicației în cadrul sistemului smart home propus.

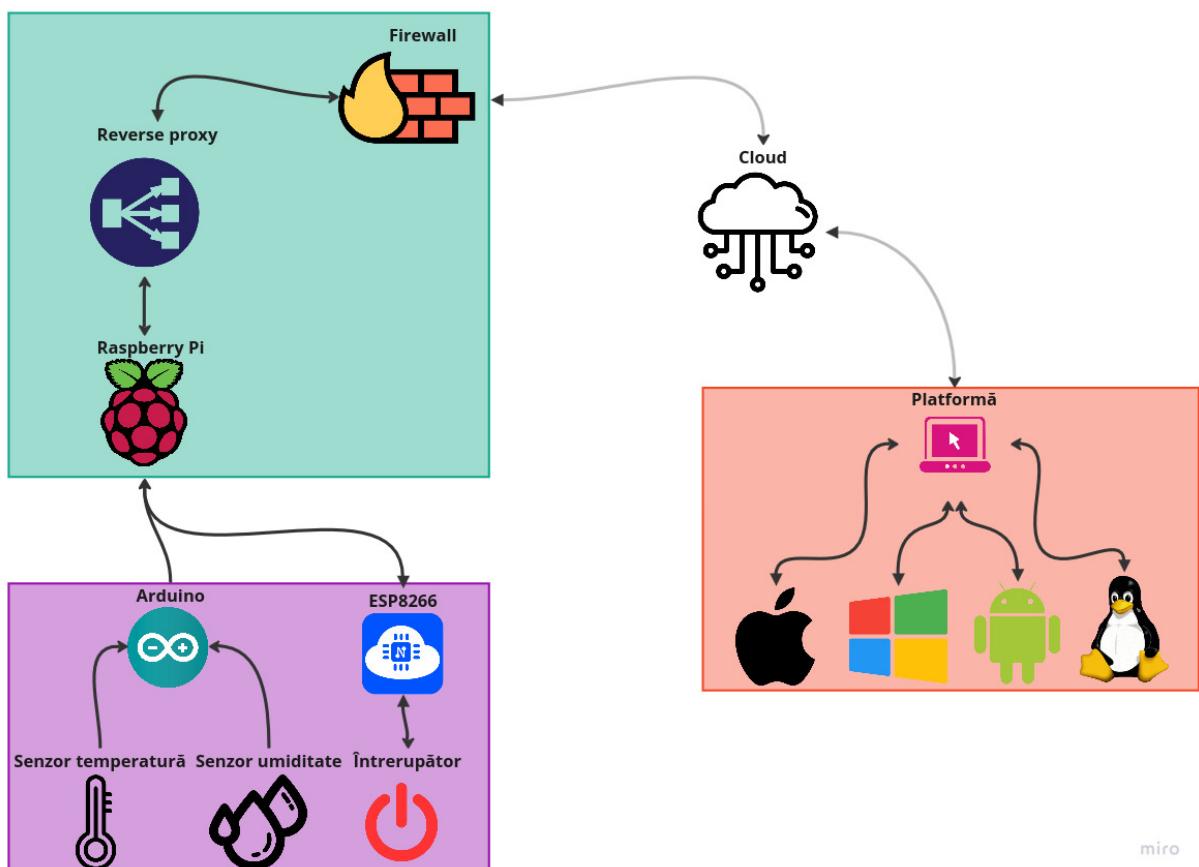


Figura 3.1: Legătura dintre componentele soluției

INSERT C4 diagram

3.1 Pagina principală

Primul punct de contact dintre un utilizator și acest proiect este aplicația de telefon (Figura 3.2). Odată lansată, va prezenta un splash screen, urmat de pagina principală. Aici se regăsesc toți senzorii conectați la sistem, împreună cu id-ul, valoarea cu timpul la care au fost înregistrate și locația din casă a senzorului.

Aceștia sunt sortați după parametrul pe care îl colectează, fie detectarea mișcării, calcularea distanței, a temperaturii sau a umidității, lucrul care se observă din titlurile situate deasupra fiecărui cartonaș. Aceste device-uri se numesc *senzori statici*, care sunt folosiți doar pentru colectare de informații. Mai există alt tip de senzor numit *dinamic* a cărui caracteristici seamănă cu cele statici, singura diferență fiind posibilitatea de comunicare bidirectională. În cazul acestui proiect, singurul senzor de acest tip va închide și deschide o lampă în circumstanțele prezentate la secțiunea proprie.

În partea de sus se află două butoane, unul de activare/ dezactivare a *modului de securitate* alături de cel de refresh care preia cele mai recente valori ale senzorilor, fiecare având o secțiune dedicată funcționalității și a modului de utilizare.

Pe post de footer sunt alte două butoane care servesc pe post de notificări atunci când valoarea unui senzor scade sub o anumită limită, respectiv cel de informații personale ale creatorului licenței.

3.1.1 Cărțile senzorilor

Cărțile reprezintă principala sursă de informații pe care un utilizator o regăsește instant când accesează aplicația. Ele oferă date importante despre senzorul respectiv într-o formă ușor de înțeles și de citit rapid.

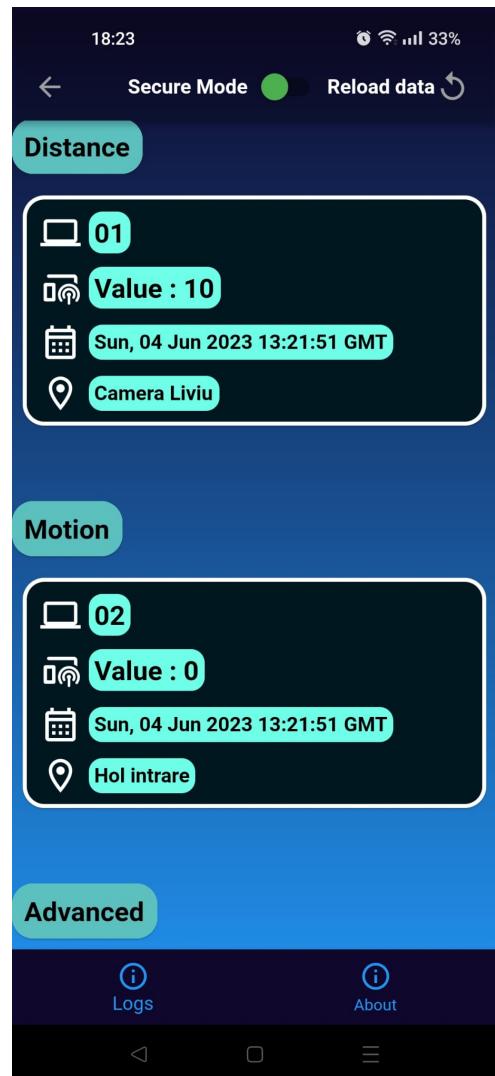


Figura 3.2: Pagina principală

În prima parte putem observa numele de identificare a senzorului. Acesta îi este asignat manual atunci când placa Arduino este programată și trebuie neapărat să fie unic în toată rețeaua.

În cazul în care există multiple dispozitive cu același ID, stația de bază va crede că a primit două pachete de la același senzor, când defapt au fost trimise de către senzori care, foarte probabil, sunt programati să colecteze date diferite și astfel vor apărea discrepanțe vizibile între valorile trimise.

După, putem observa valoarea primită de către aplicație la data și ora respectivă. Odată ce este reîmprospătată pagina, dacă device-ul este conectat la rețea, telefonul va primi cea mai recentă măsurătoare din baza de date care primește informații noi odată la două secunde.

Și nu în ultimul rând, apare și locul fizic exact al senzorului, el fiind setat de către utilizator în aplicație. Inițial, acest câmp nu este vizibil deoarece senzorii nu au atașat un loc stabil, atașarea și schimbarea lui fiind făcute de către utilizator.

Când vine vorba despre device-urile dinamice, au o interfață asemănătoare, singura diferență fiind câmpul de adresă care deține IP-ul local al senzorului.

Atunci când acest widget este apăsat, utilizatorul va intra pe pagina senzorului respectiv, detaliile căreia vor fi prezentate ADAUGA REFERINTA.

3.1.2 Locația senzorului

Cu un suport larg de diferite gesturi cu ajutorul clasei GestureDetector din Flutter, avem acces la o suiată de detectări a mișcărilor degetelor care pot executa anumite funcții. În cazul aplicației, odată ce se ține apăsat pe un cartonaș, va apărea un dialog în care putem introduce locația fizică a senzorului.

Aceasta este salvată în memoria locală a telefonului cu ajutorul bibliotecii de gestionare a bazelor de date NoSQL (non-relaționale). Am



Figura 3.3: Valorile unui modul

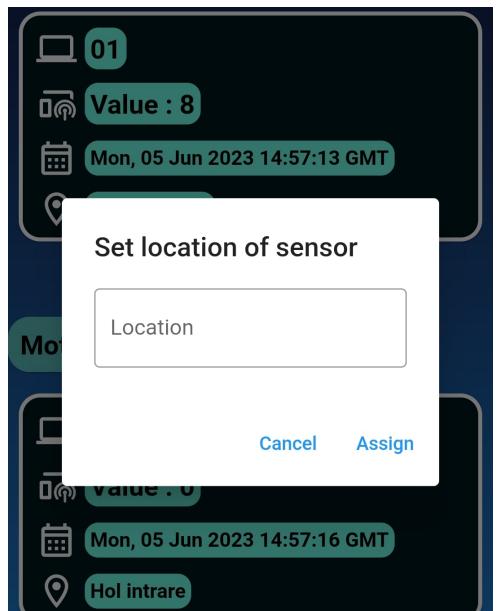


Figura 3.4: Pop-up pentru locație

ales biblioteca respectivă datorită performanței ridicate, simplitatea oferită și compatibilitatea cross-platform care este critică deoarece aplicația aceasta poate fi compilată pentru majoritatea device-urilor populare.

3.1.3 Modul de securitate

În ultimii ani, numărul de furturi din locuințe se află pe un trend descrescător, conform informațiilor oferite de către Inspectoratul General al Poliției Române (IGPR). Așadar, în anul 2019 au fost înregistrate 22341 de furturi, iar în 2020 infracțiunile au ajuns la 18060, cu 19% mai puține față de anul precedent. Chiar și în aceste circumstanțe, acest număr se datorează mentalității defensive în continuă dezvoltare și a sistemelor de alarmă care devin tot mai sofisticate. Am decis că introducerea unei astfel de abilități va contribui pozitiv la securizarea locuinței în care sistemul este instalat.

Modul de securitate se încadrează la categoria de bariere *psihice* deoarece *"descurajează infractorii sau detectează în fază incipientă o tentativă de pătrundere prin efracție (sisteme de securitate perimetrală, antiefracție, control al accesului sau sisteme de supraveghere și înregistrare video)"*¹

Este recomandat ca, atunci când utilizatorul pleacă din incinta în care este instalată soluția smart home, să activeze modul de securitate. Fiecare senzor îi poate fi atribuit o limită astfel încât orice valoare preluată care scade sub acest

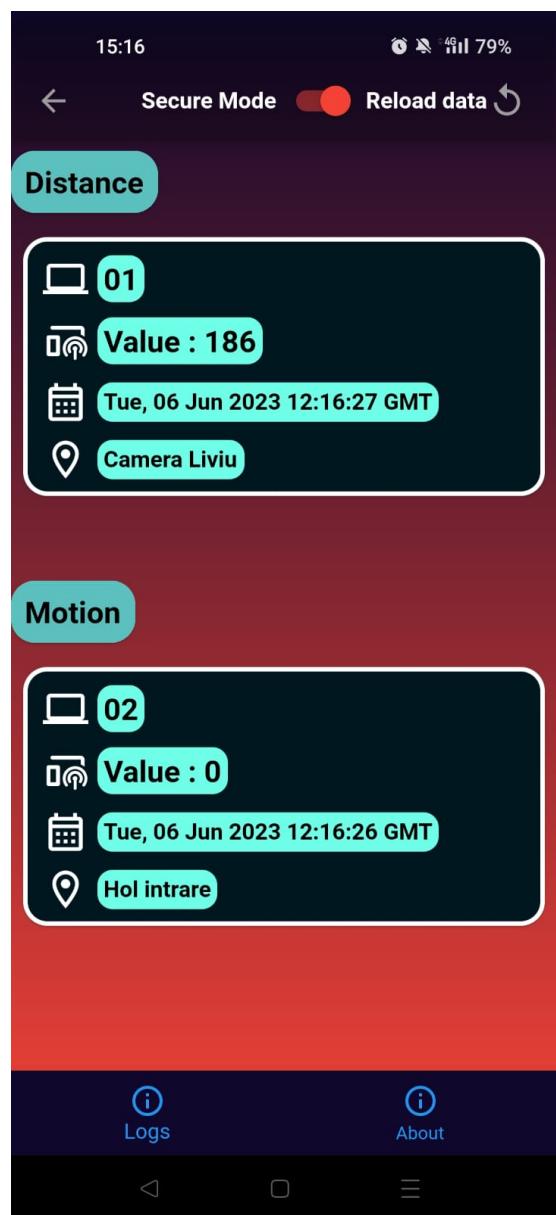


Figura 3.5: Interfață secure mode

¹<https://www.politiaromana.ro/ro/prevenire/furturi-din-locuinte>.

prag (numit și *Threshold*), va apărea sub forma unei notificări pe pagina de *Logs*.

După ce este activat, utilizatorului îi este făcut prezent acest fapt prin schimbarea culorii de background. Valoarea acestui buton este salvată în baza de date Hive, împreună cu data și ora la care a fost schimbată. Aceasta din urmă va fi folosită la filtrarea alarmelor false.

Senzorii dinamici pot fi configurați astfel încât să se folosească de această valoare și din acest motiv, la fiecare schimbare a acestui buton, valoarea va fi postată pe ruta **https://andr3w.ddns.net/secure**, urmând să fie disponibilă tuturor device-urilor pentru a o putea citi.

3.1.4 Butonul de refresh

Funcționalitatea sa este intuitivă, întrucât apăsarea lui rezultă în re-împrospătarea valorilor din aplicație cu cele mai recente citiri ale tuturor senzorilor sistemului. Deoarece tot backend-ul este hostat pe un procesor cu un singur nucleu, am decis să implementez o metodă de *caching* a datelor pentru sesiunea curentă. În acest mod salvăm serverul de la multe request-uri și economisim datele mobile.

Una din limitările sistemului este că informațiile afișate nu sunt actualizate constant. Acest lucru duce la discrepanțe între valorile pe care utilizatorul le vede și cele pe care senzorii le trimit. Vom vorbi despre aceasta în capitolul dedicat viziunii pe viitor.

3.2 Pagina senzorului static

Pe această rută se află informații și setări care contribuie la sensul de Smart home: componente ce lucrează împreună pentru a aduce utilizatorului un aport esențial de securitate.

AppBar-ul este populat cu un buton de setare a unei limite, valoare despre care am vorbit în secțiunea aceasta(ADAUGA SECUTIUNE MOD DE DECURITAET). Odată ce *threshold* este setat (Figura 3.7.b), va fi salvată în baza de date locală, astfel ea persistând chiar și când restartăm aplicația. Ca să fie vizibilă utilizatorului, pe graficul de mai jos se va trasa o linie roșie (care nu există dacă nu avem o limită). Ca utilizare, limita va fi folosită la notificările de pe pagina de *Logs* pentru a determina dacă o înregistrare trebuie arătată utilizatorului în caz că valoarea ei scade sub limita asignată.

În prim plan se află graficul care afișează evoluția ultimelor N înregistrări ale senzorului respectiv în funcție de data la care au fost colectate individual. Fiindcă sunt foarte multe puncte afișate, nu am putut include vizual și valourile lor, dar, utilizatorul o poate vedea, împreună cu data aferentă dacă apasă pe oricare punct (Figura 3.7.a). Pentru construcția sa, am decis să folosesc librăria **fl_chart** datorită popularității și a documentației excelente. Este un grafic fluid deoarece odată ce se modifică valorile afișate, motorul **fl_chart** creează o animație de tranziție de la informațiile vechi la cele noi.

Dedesubtul figurii apare un slider care setează numărul de puncte extrase de la server. A existat o problemă aici care îmi făcea un API call la fiecare schimbare de valoare, adică dacă de la 39 de puncte doresc să îmi afișeze 200, va face $200-39 = 161$ request-uri către backend. Remediul a fost ca reîmprosătarea să fie făcută în funcția **onChangeEnd** care îmi dă libertatea să aleg orice valoare din interval atât timp cât degetul încă face contact cu ecranul. Atunci când am luat degetul de pe ecran, Flutter va face request-ul, evitând astfel un semi *Denial of Service* pentru server.

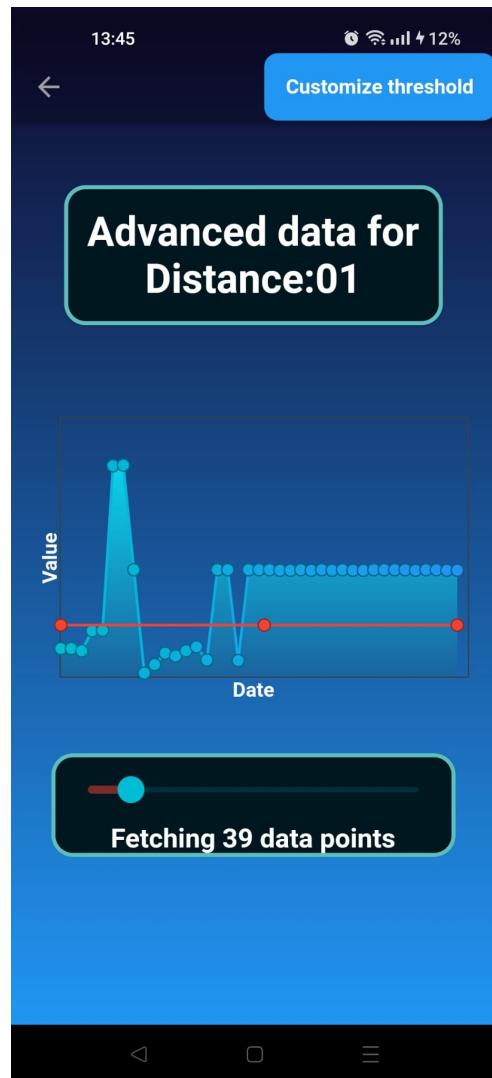
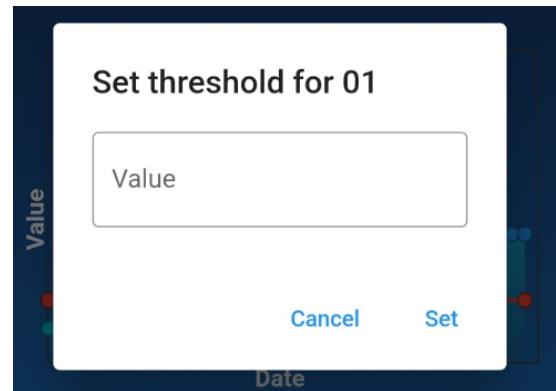


Figura 3.6: Date avansate despre senzorul 01



(a) Grafic



(b) Dialog limită

Figura 3.7

3.3 Pagina senzorului dinamic

Modulele dinamice sunt destul de diferite față de cele statice, întrucât se conectează direct la rețeaua locală de Wi-Fi (fiindu-le asignat o adresă IP) și pot primi sau trimite comenzi la server sau la alte module dinamice. M-am decis să adaug acest tip de senzori datoreită recomandării îndrumătorului licenței, domnul profesor Vidrașcu, o adiție ce se pliază perfect pe subiectul abordat în această teză: o soluție smart home.

După cum se poate observa, avem afișate datele despre senzor într-o manieră similară cu cea de la cei statici, diferență fiind IP-ul. Senzorii dinamici pot fi controlați de către utilizator și, în acest caz, este prin intermediul butoanelor ce apeleză *endpoint-urile* hostate pe ESP8266. Un endpoint reprezintă o locație specifică către care se pot face cereri HTTP (GET, POST, PUT, DELETE) pentru a accesa și manipula resursele unui serviciu. Serverul de pe ESP8266 își publică endpoint-urile la stația de bază care arată astfel: <http://192.168.0.101/turnOffLamp>.

Fiecare rută pe care o apelăm va instrui senzorul să execute diferite funcționalități care pot fi deduse din numele afișat pe buton.

IntruderLampAlert desemnează cum reacționează

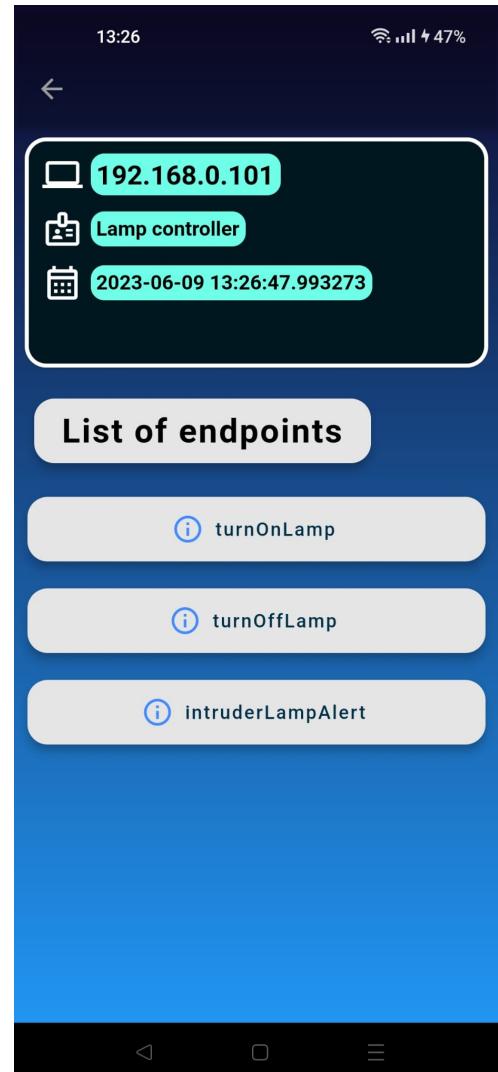


Figura 3.8: Informații și rutete senzorului dinamic

acest modul atunci când modul de securitate este activat și senzorul de mișcare emite un "1" logic.

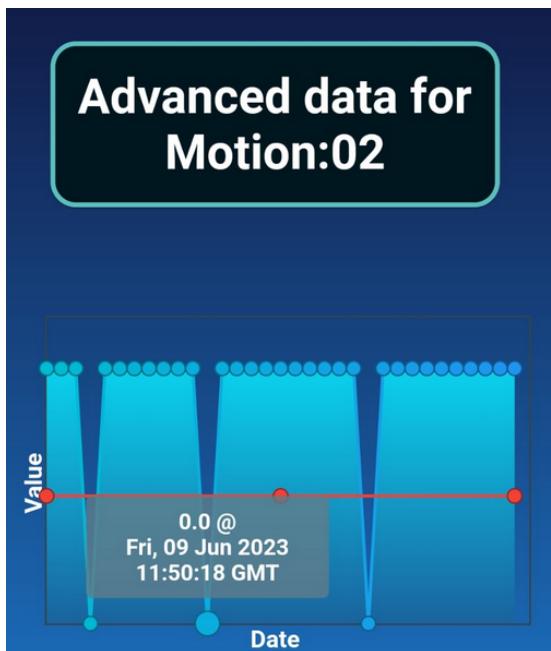
3.4 Pagina de Logs

Fiecare anomalie din sistem (orice senzor ce înregistrează o valoare care a scăzut sub limita impusă) va crea o notificare pe care utilizatorul o poate verifica în secțiunea de **Logs**. Aceasta poate fi accesată din bara de footer al meniului principal.

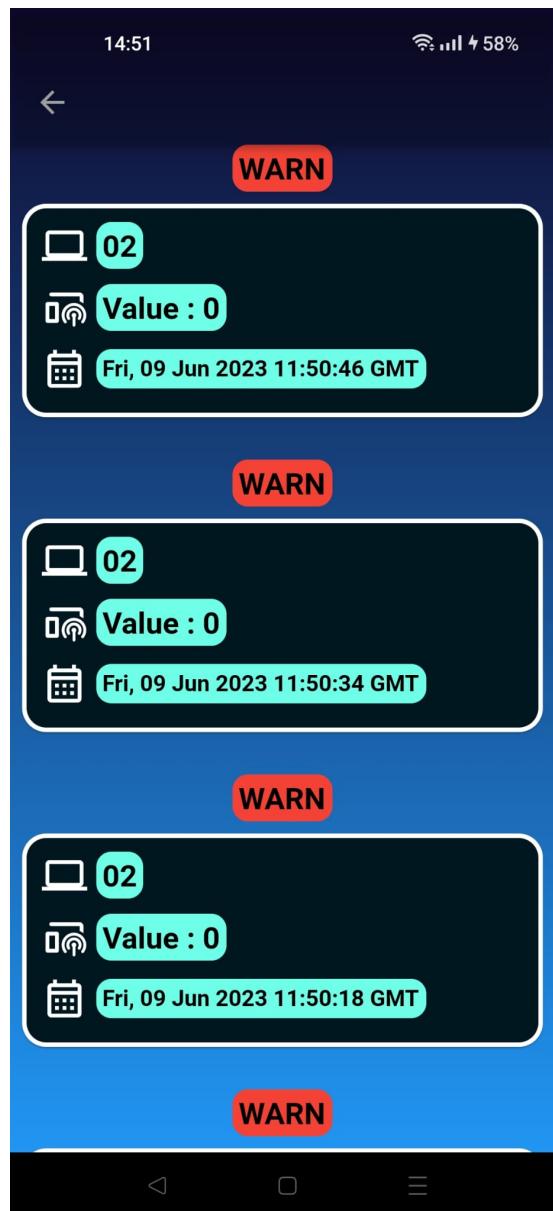
Initial, aplicația nu generează notificări dacă un senzor a înregistrat o abatere de la normal atunci când **Secure Mode** nu este activat deoarece se presupune că utilizatorul nu a plecat încă de acasă. Activarea modului de securitate va cauza afișarea anomaliei pe ecran.

Din figura 3.9.a se pot extrage anumite informații astfel: există trei puncte care se află sub bariera setată cu ajutorul butonului **Set threshold**, ceea ce înseamnă că utilizatorul și-a părăsit apartamentul, dar încă există activitate.

Aplicația va prelua ultimele înregistrări a fiecărui senzor pe care le compară cu limita (stocată în baza de date locală) și afișează cartonașele corespunzător. Se poate observa în figura 3.9.b. că ultima alertă este cea din figura anterioară.



(a) Înregistrare ce generează warning



(b) Warning generat

Figura 3.9

3.5 Pagina de genereare de rapoarte

În cazul în care utilizatorul este o victimă a unui jaf sau unei intruziuni în casă, este important să acționeze în mod corespunzător. Primul pas este să își asigure siguranța personală și să sune imediat la serviciile de urgență sau la poliție, încercând să nu intre în contact cu posibili infractori sau să manipuleze evidențele rămase la fața locului, pentru a nu distrugă potențiale probe. După anunțarea autorităților, trebuie făcută o

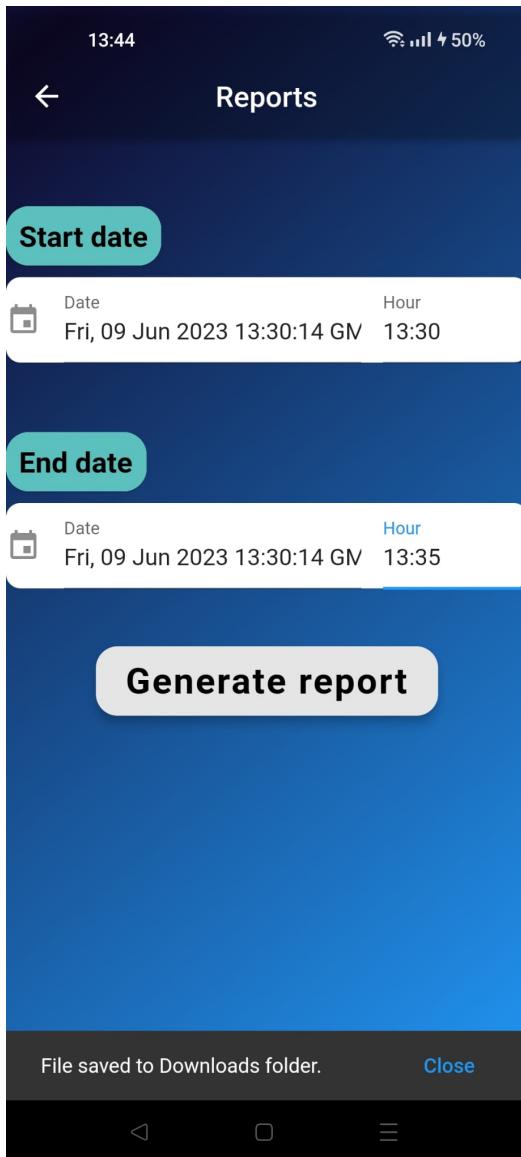
inventariere a obiectelor pierdute și să fie extrase cât mai multe informații din ziua respectivă legate de incident.

Aplicația din această soluție ajută utilizatorul în cazul unui astfel de eveniment prin colectarea tuturor datelor senzorilor din casă, expunându-le în formă de PDF gata să fie oferite autorităților. Trebuie menționat faptul că senzorii au fost plasați specific în holul de la intrarea în apartamentul meu, poziționați astfel încât orice persoană care intră va modifica starea modulelor.

Interfața (Figura 3.9.a) este alcătuită din cele două câmpuri ce permit introducerea timpului de început și cel de sfârșit. Acestea vor fi folosite de către server pentru a filtra toate înregistrările senzorilor, luându-le doar pe cele ce se încadrează în intervalul specificat. Cele două valori trebuie neapărat setate, încărcă butonul de **Generate report** nu va funcționa fiindcă nu știe unul din capetele intervalului sau chiar ambele. Odată apăsat, va trimite un POST request la stația de bază care îi va răspunde cu datele aferente. Utilizatorul este notificat de terminarea procesului printr-un pop-up în josul ecranului ce oferă și folderul în care a fost salvat.

Cu ajutorul librăriei **pdf** din cadrul SDK-ului Flutter, am reușit generarea documentului (Figura 3.9.b). Pe post de header, am specificat intervalul din care au fost extrase informațiile senzorilor și data la care a fost creat acest document. Footer-ul conține numele aplicației și detinătorul drepturilor licenței. În centrul atenției este un tabel ce conține, în ordine, numărul curent al înregistrării, adresa, data, ID-ul, tipul și valoarea senzorului. Acesta este extins pe multiple foi, cifra din partea de jos al ecranului desemnând numărul curent al paginii. Pentru raportul acesta avem un număr de 300 de înregistrări preluate în decursul a 5 minute.

Datele acestui document sunt valoroase și pot ajuta foarte mult investigatorii, crescând astfel șansele de a prinde intrușii. Asigurând proprietarii ce folosesc acest sistem cu un raport plin de informații, soluția de smart home propusă joacă un rol important în creșterea securității utilizatorului.



(a) Interfața paginii

Data recorded between
Fri, 09 Jun 2023 10:30:00 GMT and Fri, 09 Jun 2023 10:35:00 GMT.
Report has been generated on: Fri, 09 Jun 2023

Count	Address	Date	ID	Type	Value
141	02	Fri, 09 Jun 2023 10:32:39 GMT	63188	Motion	0
142	01	Fri, 09 Jun 2023 10:32:39 GMT	63187	Distance	9
143	02	Fri, 09 Jun 2023 10:32:37 GMT	63186	Motion	0
144	01	Fri, 09 Jun 2023 10:32:37 GMT	63185	Distance	8
145	02	Fri, 09 Jun 2023 10:32:35 GMT	63184	Motion	0
146	01	Fri, 09 Jun 2023 10:32:35 GMT	63183	Distance	8
147	02	Fri, 09 Jun 2023 10:32:33 GMT	63182	Motion	0
148	01	Fri, 09 Jun 2023 10:32:33 GMT	63181	Distance	8
149	02	Fri, 09 Jun 2023 10:32:31 GMT	63180	Motion	0
150	01	Fri, 09 Jun 2023 10:32:31 GMT	63179	Distance	8
151	02	Fri, 09 Jun 2023 10:32:29 GMT	63178	Motion	0
152	01	Fri, 09 Jun 2023 10:32:29 GMT	63177	Distance	8
153	02	Fri, 09 Jun 2023 10:32:27 GMT	63176	Motion	0
154	01	Fri, 09 Jun 2023 10:32:27 GMT	63175	Distance	8
155	02	Fri, 09 Jun 2023 10:32:25 GMT	63174	Motion	0
156	01	Fri, 09 Jun 2023 10:32:25 GMT	63173	Distance	8
157	02	Fri, 09 Jun 2023 10:32:23 GMT	63172	Motion	0
158	01	Fri, 09 Jun 2023 10:32:23 GMT	63171	Distance	8
159	02	Fri, 09 Jun 2023 10:32:21 GMT	63170	Motion	0
160	01	Fri, 09 Jun 2023 10:32:21 GMT	63169	Distance	8
161	02	Fri, 09 Jun 2023 10:32:19 GMT	63168	Motion	0
162	01	Fri, 09 Jun 2023 10:32:19 GMT	63167	Distance	8
163	02	Fri, 09 Jun 2023 10:32:17 GMT	63166	Motion	0
164	01	Fri, 09 Jun 2023 10:32:17 GMT	63165	Distance	8
165	02	Fri, 09 Jun 2023 10:32:15 GMT	63164	Motion	0
166	01	Fri, 09 Jun 2023 10:32:15 GMT	63163	Distance	8
167	02	Fri, 09 Jun 2023 10:32:13 GMT	63162	Motion	0
168	01	Fri, 09 Jun 2023 10:32:13 GMT	63161	Distance	8

Licensed to: Fil_Report generated by:
Smart Home App 2023

(b) PDF-ul generat

Figura 3.10

3.6 Senzori statici

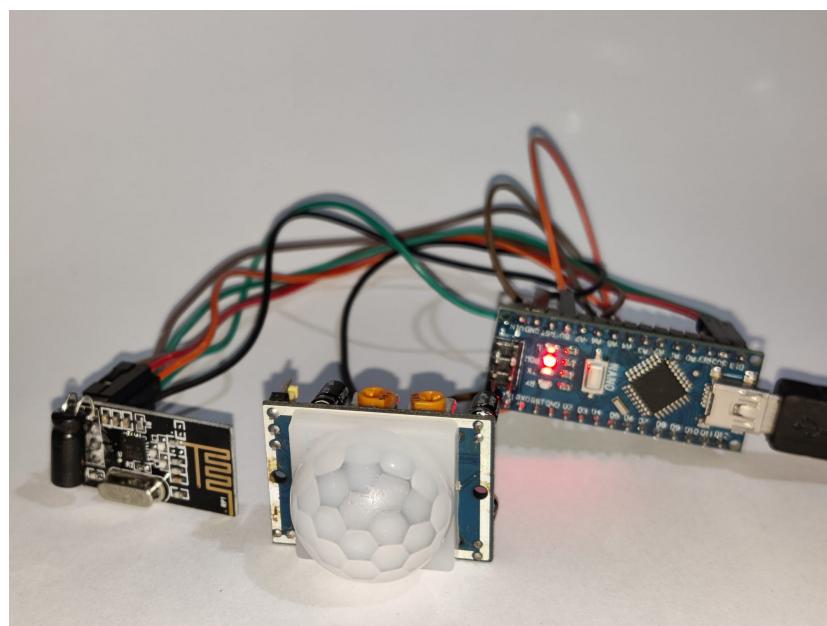
Acest tip de device-uri stă la baza metodei de colectare a informațiilor despre casa respectivă și cea de generare a unui raport cu toate valorile din baza de date în intervalul cerut.

Figura 3.3.a ne prezintă cum a fost creat un senzor static: un Arduino Nano care este cablat la senzorul de detectare a mișcării și modulul de transmitere a datelor.

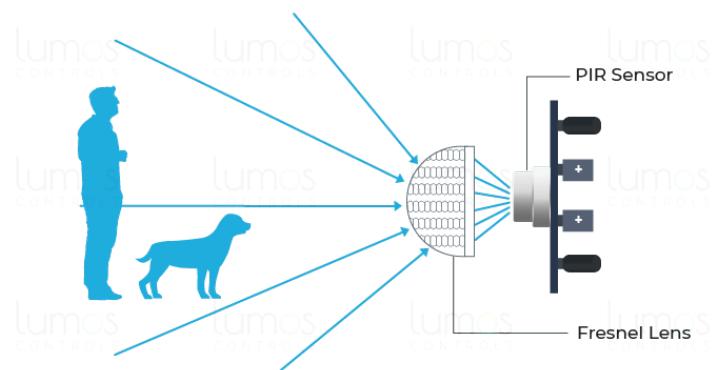
Senzorul are 3 pini care trebuie conectați la placa de dezvoltare: Ground (împământare), Vcc (+5V) și Out. Ultimul pin emite un 0 logic dacă nu a detectat niciun obiect și 1 logic dacă a detectat o persoană sau un animal. Acest lucru este posibil datorită faptului

că este emisă căldură în formă de radiații infra-roșii care alimentează componenta *piroelectrică* (Figura 3.3.b) amplasată dedesubtul unei lentile Fresnel ce ajută focusarea acestor raze pe lentila senzorului. Modulul are încorporat două potențiometre: unul pentru reglarea razei de acțiune (până în 5 metri), iar celălalt pentru reglarea timpului de schimbare a valorii detectate (între 0.3 - 300 secunde).

Odată alimentat la 5V, microcontroller-ul începe colectarea de date odată la două secunde. După ce acest interval a trecut, va fi creat un pachet ce conține ID-ul și tipul modulului, valoarea și data la care a fost făcută măsurătoarea.



(a) Arduino Nano + senzor de mișcare + modul transmisie



(b) Componență piroelectrică

Figura 3.11

Transmiterea și primirea acestor pachete va fi detaliată în secțiunea despre placă NRF24L01 unde vom vorbi de banda 2.4GHz, structura unui pachet de date și RF24Network.

3.7 Senzori dinamici

Senzorii dinamici dețin o complexitate mai ridicată față de tipul anterior prezentat datorită faptului că pot trimite și primi date simultan de la server, făcându-i să pară că folosesc o conexiune *full duplex*. Din cauza faptului că legătura dintre server și senzor se face prin intermediul protocolului Wifi, înseamnă că această legătură este mai degrabă *half duplex*, schimbul de date realizându-se prin alternarea transmițătorului și receptorului, iar dispozitivele trebuie să respecte un anumit protocol de acces pentru evitarea coliziunilor și interferențelor cu alte dispozitive. CITAT

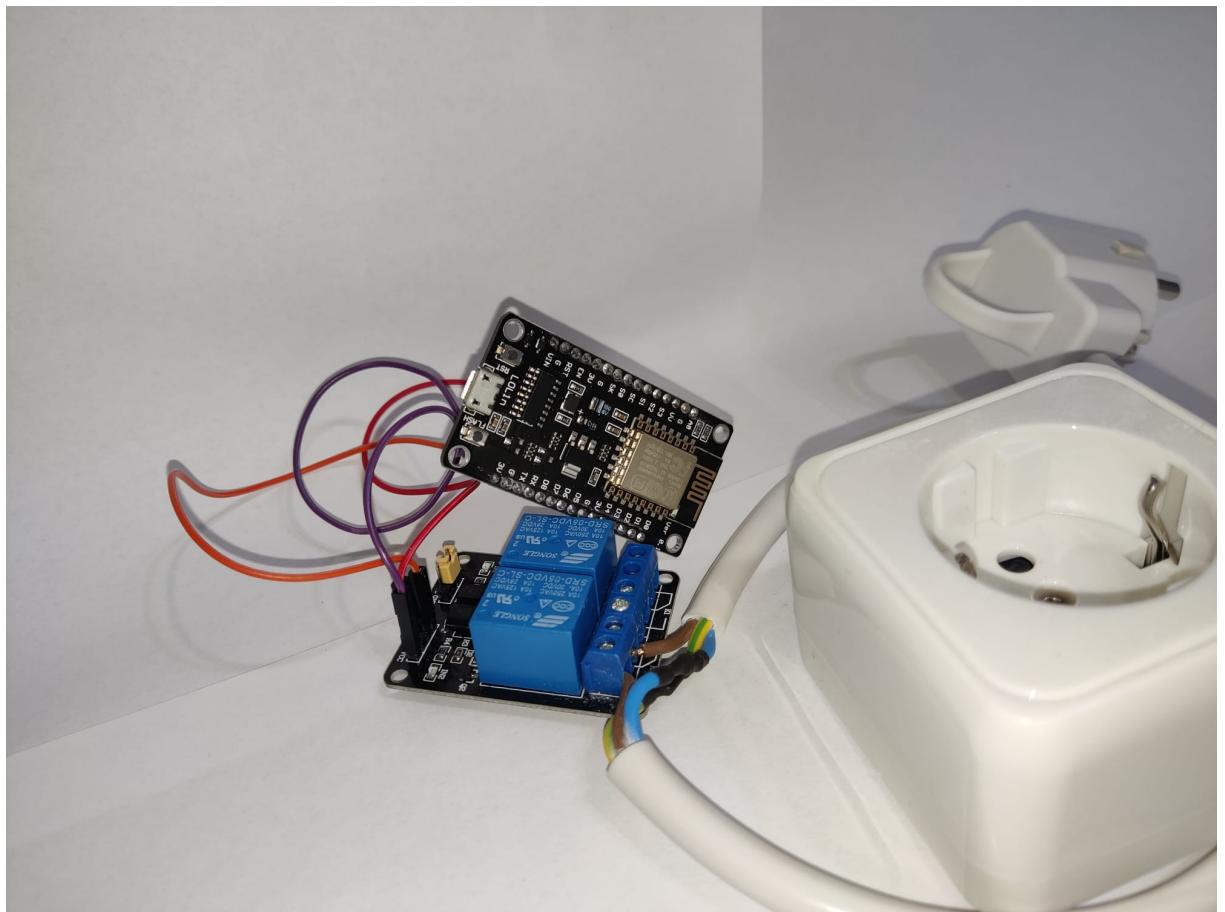


Figura 3.12: ESP8266 + modul releu alimentat la 220V

Concluzii

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nunc mattis enim ut tellus elementum sagittis vitae et. Placerat in egestas erat imperdiet sed euismod. Urna id volutpat lacus laoreet non curabitur gravida. Blandit turpis cursus in hac habitasse platea. Eget nunc lobortis mattis aliquam faucibus. Est pellentesque elit ullamcorper dignissim cras tincidunt lobortis feugiat. Viverra maecenas accumsan lacus vel facilisis volutpat est. Non odio euismod lacinia at quis risus sed vulputate odio. Consequat ac felis donec et odio pellentesque diam volutpat commodo. Etiam sit amet nisl purus in. Tortor condimentum lacinia quis vel eros donec. Phasellus egestas tellus rutrum tellus pellentesque eu tincidunt. Aliquam id diam maecenas ultricies mi eget mauris pharetra. Enim eu turpis egestas pretium.

Bibliografie

- Author1, *Book1*, 2018
- Ghid Licență
- Creating the ESP8266 server
- Raspberry Pi Foundation
- IFTTT
- Raspberry Pi Zero W
- Raspberry Pi Zero W (1)
- Raspberry Pi Zero W (2)
- Arduino Nano
- PIR Sensor
- Senzor de temperatură și umiditate
- Funcționalitate senzor PIR
- Funcționalitate senzor PIR (2)
- Prevenire furturi