

MINISTERUL EDUCAȚIEI



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Sistem de automatizare și monitorizare a locuințelor

LUCRARE DE LICENȚĂ

Absolvent: **Alexandru Pavel**

Conducător științific: **Dr. Ing. Gabriel Dragomir-Loga**

2021

Cuprins

Capitolul 1 Introducere - Contextul proiectului	1
1.1 Contextul proiectului	1
1.2 Structura lucrării	2
Capitolul 2 Obiectivele Proiectului	3
2.1 Obiective	3
2.2 Specificații	4
Capitolul 3 Studiu Bibliografic	6
3.1 Introducere în domeniul IoT	6
3.2 Sistemele Smart Home	8
3.3 Automatizarea imobilelor cu Arduino	10
Capitolul 4 Analiză și Fundamentare Teoretică	11
4.1 Scopul proiectului	11
4.2 Cerințele sistemului	11
4.2.1 Cerințe funcționale	11
4.2.2 Cerințe non-funcționale	12
4.3 Perspectiva tehnologică	15
4.3.1 Interfețe de comunicare serială	15
4.3.2 Componente hardware utilizate	17
4.3.3 Tehnologii utilizate	26
Capitolul 5 Proiectare de Detaliu și Implementare	28
5.1 Arhitectura sistemului	28
5.2 Descrierea vizuală a sistemului	30
5.2.1 Diagrama cazurilor de utilizare	30
5.2.2 Diagrame de secvență	31
5.3 Implementarea sistemului local	32
5.3.1 Componențele necesare implementării	32
5.3.2 Schema electrică a părții locale	33
5.3.3 Librării utilizate	34

5.3.4	Meniul sistemului	35
5.3.5	Controlul accesului în locuință	36
5.3.6	Funcția de termostat	36
5.3.7	Funcția de irigare	37
5.3.8	Detectia factorilor de risc	37
5.3.9	Comunicarea cu serviciul de Cloud	38
5.4	Implementarea aplicației mobile	40
5.4.1	Diagrama de clase	40
5.4.2	Structura bazei de date	41
5.4.3	Librării utilizate	42
5.4.4	Logarea în contul de utilizator	43
5.4.5	Monitorizarea parametrilor	44
5.4.6	Trimitearea de comenzi	45
5.4.7	Alerte de urgență	46
Capitolul 6	Testare și Validare	47
6.1	Testarea sistemului local	47
6.2	Testarea aplicației mobile	48
Capitolul 7	Manual de Instalare și Utilizare	50
7.1	Ghid de instalare	50
7.2	Manual de utilizare	52
7.2.1	Utilizarea sistemului local	52
7.2.2	Utilizarea aplicației mobile	55
Capitolul 8	Concluzii	59
8.1	Realizări și analiza rezultatelor obținute	59
8.2	Dezvoltări și îmbunătățiri ulterioare	60
Bibliografie		62

Capitolul 1

Introducere - Contextul proiectului

1.1 Contextul proiectului

Siguranța și monitorizarea locuințelor a devenit un aspect important în viața proprietarilor de imobile, avansul tehnologic actual permitând realizarea acestor două aspecte de la distanță, în timp real și fără costuri majore. Astfel au apărut diverse variante de sisteme de domotizare (automatizare la domiciliu) a locuințelor numite și sisteme Smart Home, cererea acestora fiind în creștere continuă pe piață. Aceste sisteme fac parte din ecosistemul Internet of things, cunoscut și după numele de IoT.

Internet of things sau IoT este un concept care presupune formarea unei rețele de dispozitive interconectate prin intermediul internetului, acestea putând comunica între ele cu ajutorul diferitor protocoale[1]. Apariția acestui domeniu a facilitat dezvoltarea unor subdomenii noi, automatizarea putând fi adusă cu usurință chiar la client acasă. Ca și vulnerabilitate prezentă în cadrul majorității sistemelor din cadrul acestui domeniu se remarcă securitatea, dispozitivele fiind predispușe la atacuri cibernetice și colectare de date.

Conceptul de Smart Home are la bază interconectarea unui sistem local aflat în interiorul locuinței, format din diferiți senzori (temperatură, umiditate, lumină, mișcare, etc.) și actuatoare (iale electrice, robinete electrici, etc.) care sunt conectați în permanentă la un server local. Acesta comunică în permanentă cu diferite servicii web pentru transmiterea în timp real a datelor procesate și primirea de date noi[2].

Proiectul de față aduce o soluție facilă pentru implementarea unui sistem complet de automatizare și monitorizare al imobilelor, oferind toate facilitățile întâlnite la competiție prin intermediul unei aplicații mobile dedicate. Elaborarea proiectului a fost facută în concordanță cu principiile domeniului IoT, fiind necesară conexiunea permanentă la internet a sistemului local pentru o funcționare corectă. Sistemul a fost structurat pe trei nivele, și anume partea locală, partea de Cloud și partea de aplicație destinată utilizatorului. Delimitarea pe nivele a soluției aduce avantaje precum diagnosticarea mai ușoară a diferitor bug-uri apărute pe parcursul utilizării, dar și adăugarea ușoară de funcționalități noi.

1.2 Structura lucrării

Documentul este strucutrat pe 8 capitole care conțin multiple subcapitole, urmărind procesul de documentare și implementare al unui sistem complet de automatizare și monitorizare al locuințelor.

Primul capitol prezintă contextul realizării proiectului, fiind surprins aspecte precum motivația din spatele acestuia în concordanță cu domeniul de activitate în care se încadrează. Este prezentată și structura generală a proiectului, alături de alte caracteristici specifice.

Al doilea capitol prezintă tematica concretă a proiectului și obiectivele necesare implementării. Sunt elaborate specificațiile proiectului și sunt prezentate etapele proiectării cu modul aferent de funcționare dorit.

Al treilea capitol reprezintă studiul bibliografic necesar pentru elaborarea proiectului. Astfel sunt prezentate studiile și articolele pe care sunt bazate implementările celor trei părți ale sistemului, și anume partea locală, partea de Cloud și partea aplicației mobile.

Al patrulea capitol cuprinde noțiunile teoretice care stau la baza implementării proiectului, acestea fiind elaborate în detaliu. Soluția descrisă în această secțiune este prezentată dintr-un punct de vedere strict teoretic, aspectele definite în această etapă fiind necesare în continuarea implementării.

Al cincilea capitol reprezintă implementarea propriu-zisă a proiectului, fiind prezentată structura modulară a proiectului și algoritmica din spatele diferitelor părți din cadrul implementării. Etapa curentă este bazată pe fundamentarea teoretică realizată în cadrul capitolului anterior.

Al șaselea capitol reprezintă etapa de testare din cadrul dezvoltării proiectului. Sunt comparate funcționalitățile obținute și acuratețea acestora cu cele ale competiției de pe piața de sisteme Smart Home, validarea facându-se în concordanță cu sistemele deja existente.

Al șaptelea capitol cuprinde manualul de instalare și utilizare al proiectului, fiind prezentate resursele de care utilizatorul trebuie să dispună pentru o experiență completă. De asemenea sunt menționati și pașii de operare ai sistemului pentru ca utilizarea să fie una fluidă, cu rezultatele așteptate.

Ultimul capitol prezintă concluziile rezultate în urma elaborării proiectului. Sunt prezentate și posibile dezvoltări ulterioare, destinate imbunătățirii sistemului prin adăugarea de module sau utilizarea unor tehnologii mai noi.

Capitolul 2

Obiectivele Proiectului

2.1 Obiective

Proiectul elaborează un sistem complet de automatizare și monitorizare al locuințelor. Obiectivul principal al acestuia este de a oferi proprietarilor de imobile un sistem fiabil pentru vizualizarea în timp real a parametrilor din cadrul rezidenței, oferind posibilitatea de trimisire a comenziilor de la distanță.

Sistemul este bazat pe trei nivele structurale, fiecare nivel având obiective specifice pentru o funcționare corectă. Funcționarea defectuoasă a unui nivel duce la funcționarea defectuoasă a întregului sistem, cele trei nivele având astfel o relație de interdependentă între ele.

Nivelul local este format din senzori, actuatoare și server, acestea fiind esențiale pentru monitorizarea parametrilor din imobil și realizarea anumitor comenzi de la distanță.

Nivelul de Cloud este necesar pentru realizarea conexiunii dintre partea locală și aplicația mobilă destinată utilizatorului, implicând un nivel ridicat de securitate pe baza serviciilor Google, datele fiind transmise cu ajutorul unei baze de date de tip Real-time.

Ultimul nivel, și anume nivelul utilizator sau al aplicației mobile, reprezintă portalul de acces dintre utilizator și sistemul Smart Home, oferind toate funcționalitățile specifice în cadrul unei interfețe intuitive.

Astfel, în urma analizei celor trei nivele care intră în alcătuirea proiectului se pot regăsi urmatoarele obiective:

- Funcționarea și prelucrarea corectă a senzorilor/actuatoarelor locali;
- Conexiunea stabilă la internet a sistemului local cu ajutorul modulului de wireless programat;
- Funcționarea corectă a serverului local și a tuturor componentelor aferente în urma interconectării;
- Funcționarea corectă a serviciului de Cloud, fără pierderi de date, cu transmitere duală;

- Securitatea serviciului de Cloud pentru combaterea furtului de date;
- Interfața intuitivă și fluidă a aplicației mobile;
- Funcționarea corectă a aplicației mobile din punct de vedere al funcționalităților specifice;
- Întârziere de transfer minimă a datelor între cele trei nivele;
- Funcționarea corectă a tuturor celor trei nivele, în mod simultan, oferind un sistem de Smart Home complet.

2.2 Specificații

În urma menționării obiectivelor stabilite în cadrul implementării proiectului rezultă specificațiile aferente acestuia, și anume:

- Sistemul local dispune de senzori integrați și module precum un cititor de carduri RFID, dar și de un modul dedicat de relee pentru conectarea de dispozitive externe/actuatorare. Acestea lucrează în mod paralel cu ajutorul microcontrollerului integrat, datele din cadrul imobilului fiind astfel prelucrate în timp real. Comenzile primite sunt interpretate, modulul de relee fiind acționat pentru controlul dispozitivelor dorite/actuatorelor.
- Detectiona elementelor care implică pericol în cadrul imobilului, precum flăcări sau surgeri de gaz metan, declanșează alerte de severitate ridicată. Fiecare alertă activează o alarmă de urgență emisă de serverul local, dar și notificări de severitate ridicată pe partea aplicației mobile.
- Utilizatorul are la dispoziție un ecran lcd integrat pe partea de server cu meniuri specifice, acestea fiind navigabile cu ajutorul tastaturii integrate. Meniurile oferă informații precum temperatura actuală, statusul diferitor dispozitive din imobil sau alarme de urgență. Utilizatorul are astfel acces la funcții precum termostat și armarea securității și de pe partea locală a sistemului, fiind însă o versiunea mai restrânsă decât cea oferită de aplicația mobilă dedicată.
- Sistemul asigură securitatea imobilului prin controlarea unei iale electrice, cu închidere automată temporizată. Aceasta poate fi acționată cu ajutorul unui card RFID sau din interiorul aplicației mobile, prin apăsarea unui simplu buton. De asemenea un senzor de detecție al mișcării este integrat în cadrul parții locale, fiind necesara armarea acestuia din cadrul aplicației/meniului navigabil de către utilizator.
- Conexiunea sistemului la internet este asigurată de un modul wireless dedicat, programat pentru a trimite și a primi datele într-un anumit format. Acesta necesită

conexiunea permanentă la o rețea locală pentru o funcționare corectă a întregului sistem, deconectarea serverului local de la internet rezultând în neactualizarea datelor din aplicația mobilă și imposibilitatea de a trimite comenzi de la distanță.

- Transmiterea datelor între partea locală și aplicația mobilă se realizează cu ajutorul serviciilor de Cloud, fiind folosită o bază de date de tipul Real-time. Există două canale de comunicare diferite pentru fiecare direcție de transfer, datele primite de la serverul local fiind inserate ca și intrări în baza de date pentru a putea fi interpretate în viitor.
- Logarea în cadrul aplicației mobile se realizează cu ușurință prin cadrul contului de Google legat la baza de date de tipul Real-time aferentă sistemului, nefiind necesară crearea de conturi noi.
- Aplicația mobilă oferă utilizatorului acces complet și în timp real asupra datelor colectate din imobil. Astfel sunt afișate informații precum umiditatea, dar și starea diferitor dispozitive precum sistemul de climatizare. Aplicația mobilă oferă și posibilitatea de comenzi de la distanță, precum setarea temperaturii dorite în imobil, deblocarea șalei principale și programarea duratei de irigare a plantelor.
- Aplicația destinată utilizatorului nu are un mod de navigare secvențial, toate funcționalitățile fiind regăsite în cadrul aceleasi pagini derulabile, fiind astfel posibilă trimiterea mai multor comenzi simultan spre partea locală, fără a fi afectată primirea de date în timp real.

Capitolul 3

Studiu Bibliografic

3.1 Introducere în domeniul IoT

IoT sau Internet of things este o paradigmă tehnologică care presupune realizarea comunicării între diverse dispozitive, senzori și actuatori cu ajutorul rețelelor de internet și a protocolelor de comunicare specifice. Este un concept inovativ în majoritatea domeniilor de activitate, oferind facilități atât pentru cadrele private cât și pentru instituțiile publice. Evoluția din ultimii ani a domeniului IoT a crescut semnificativ nivelul de trai și calitatea serviciilor oferte de diverse câmpuri de activitate.

Posibilitatea de integrare a diferitor sub-sisteme în cadrul același sistem este o caracteristică importantă a IoT-ului, făcând posibil controlul și monitorizarea în timp real de la distanță. Caracterul volatil este exprimat și prin posibilitatea interconectării de tehnologii inițial incompatibile, cum ar fi framework-uri, senzori și dispozitive inteligente. Se folosesc protocole de comunicare cu API-uri configurabile, destinate integrării ulterioare a sub-sistemelor noi.[?].

Progresul recent în câmpul calculatoarelor cuantice și al nanotehnologiei oferă de asemenea un avantaj în cadrul domeniului IoT, fiind atinse viteze de transfer și de procesare a datelor neîntâlnite în anii anteriori[3]. Stabilitatea crescută a sistemelor de Cloud actuale și securitatea complexă oferă oricărui dezvoltator posibilitatea de transmitere a datelor în timp real, cu delay-uri nesemnificative utilizării. Nanotehnologia conferă de asemenea un aspect portabil sistemelor de IoT, făcând posibilă realizarea anumitor acțiuni fără nevoie unui specialist în domeniu. Se impun astfel beneficii precum reducerea consumului de energie prin utilizarea dispozitivelor nano, acuratețea de monitorizare crescută și fiabilitatea ridicată datorită dimensiunilor reduse.

Piața domeniului IoT se extinde progresiv, cuprinzând domenii precum interconectarea utilajelor, orașe inteligente, economisirea energiei, automobile cu sisteme inteligente, agricultură și vânzări. În Figura 3.1 sunt prezentate principalele domenii de activitate în care parada actuală este regăsită cu succes, oferind o statistică bazată pe cota globală a fiecărui domeniu. Fiind încă la început, dezvoltatorii prezintă încă un număr limitat de proiecte în desfășurare, numărul aproximativ al acestora fiind cuprins undeva între 7000 și

3.1. INTRODUCERE ÎN DOMENIUL IOT

7

10000 la nivel global. Acest aspect reprezintă un impuls avantajos companiilor inovatoare, apărând periodic sisteme pentru îmbunătățirea considerabilă a nivelului de trai.[4].

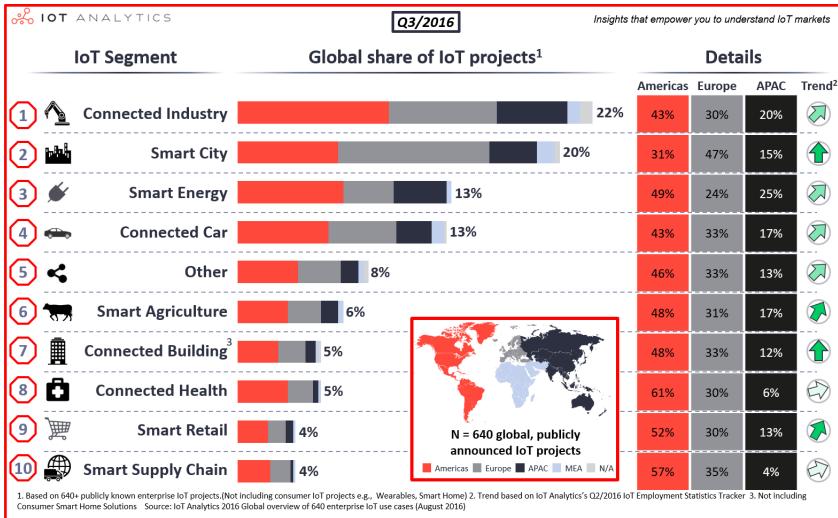


Figura 3.1: Statisitca globală a domeniilor din câmpul paradigmelor IoT[4]

Arhitectura generală a sistemelor IoT este formată din aplicațiile IoT, serverele IoT, gateway-urile IoT și dispozitivele monitorizate și controlate. Sistemul este realizat cu ajutorul aplicațiilor, care comunică în permanență cu serverele IoT. Gateway-urile sunt responsabile cu formarea conexiunii dintre serviciile de Cloud/server și dispozitivele monitorizare/controlate. Astfel se realizează un ecosistem care funcționează în timp real, având un caracter scalabil și distribuit.

Sistemele IoT trebuie să fie proiectate cu obiectivul de a putea uni funcționalitățile și cerințele mai multor domenii, integrând mai multe sisteme cu o menenanță usoară și scalabilitate ridicată, datele fiind analizate și stocate în cantitate mare. De asemenea este necesară și posibilitatea de extindere ulterioară a funcționalităților sistemului, aspect care afectează modul de proiectare pe tot parcursul dezvoltării[?].

Principalele dezavantaje actuale ale sistemelor din cadrul domeniului IoT sunt securitatea, interconectarea, fiabilitatea și calitatea serviciilor. Securitatea și confidențialitatea datelor sunt două aspecte cruciale în cadrul oricărui sistem informatic, atacurile cibernetice din ultimul deceniu fiind direcționate în mare parte către sistemele IoT, fie acestea utilizate de corporații sau de spații private. Astfel se caută permanent măsuri de combatere a acțiunilor malicioase asupra securității, conceptul principal urmărit fiind o rețea securizată și stabilă accesată doar în urma autentificării. Interconectarea sau funcționarea în paralel a diferitelor sub-sisteme este de asemenea un aspect care poate ridica probleme în momentul intreruperii conexiunii în cadrul sistemului, fiind utile canalele de comunicare individuale de backup. Fiabilitatea și calitatea serviciilor sunt de asemenea două aspecte importante, fiind cele mai discutate în cadrul experienței de utilizare. Pentru evitarea posibilelor defecțiuni se urmărește utilizarea standardelor specifice pentru toate dispozi-

tivele care intră în alcătuirea unui sistem, calitatea serviciilor fiind mereu îmbunătățită în raport cu evaluarea sistemului de către utilizatori[3].

3.2 Sistemele Smart Home

Sistemele Smart Home fac parte din cadrul paradigmelor de dezvoltare IoT sau Internet of things, fiind utilizate pentru realizarea procesului de automatizare și monitorizare al imobilelor. Sistemele de acest tip oferă o interfață intuitivă pentru navigarea funcționalităților, oferind utilizatorului confort, siguranță și eficiență. Imobilele în cadrul cărora sunt integrate asemenea sisteme sunt benefice mediului înconjurător, reducând semnificativ consumul de resurse și de materie reziduală[2]. Structura generală a unui sistem de monitorizare și automatizare a locuințelor este prezentată în Figura 3.2.

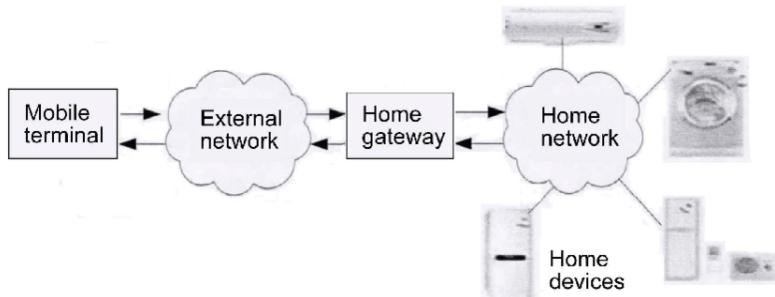


Figura 3.2: Structura generală a unui sistem Smart Home¹

În cadrul sistemelor de monitorizare și automatizare a locuințelor se regăsesc un set de funcționalități standard, acestea fiind: interfațarea comunicării dintre utilizator și sistemul local din cadrul imobilului, monitorizarea în timp real a diferitor parametrii, acțiuni remote sau de la distanță cu desfășurare în timp real și alerte de urgență în cazul posibilelor detecții. Utilizatorii sistemelor de Smart Home pun accentul în mare parte pe părțile aferente siguranței și alertelor de urgență, îmbunătățirea acestora fiind crucială pentru fiecare producător din domeniul[5].

Structura internă a unui sistem de monitorizare urmează de asemenea un format standardizat. Acesta este compus dintr-o parte locală care necesită amplasare și montare corespunzătoare, o parte de Cloud pentru comunicare și partea aplicației mobile. Partea locală este de obicei formată dintr-un hub local precum Echo Dot oferit de cei de la Amazon, hub care rulează un software dedicat pentru realizarea funcționalităților de monitorizare și automatizare. La acesta sunt conectate în mod direct diverse module periferice, cum ar fi senzori, actuatori sau sisteme controlate. Interfața de comunicare este standardizată, oferind astfel un caracter de Plug and Play. Hub-ul local necesită conexiunea permanentă la o rețea de internet pentru a satisface paradigma IoT și pentru a funcționa corespunzător.

¹https://www.researchgate.net/figure/Structure-of-smart-home-system_fig1_292343353

3.2. SISTEMELE SMART HOME

9

Comunicarea dintre partea locală și aplicația mobilă dedicată se realizează cu ajutorul serviciilor de Cloud, fiind folosite protocoale de comunicare specifice fiecărui producător. Aplicația mobilă integrează toate funcționalitățile sistemului Smart Home, oferind posibilitatea de utilizare de la distanță, în timp real. Funcționarea corectă a sistemului necesită conexiunea ambelor părți terminale la internet, și anume a hub-ului local și a aplicației utilizatorului.[2].

Pe piață sistemelor de Smart Home se remarcă 4 sisteme predominante, și anume Amazon Alexa, Google Assistant, Wink Hub și Samsung SmartThings Hub. Fiecare dintre sistemele menționate prezintă anumite avantaje și anumite părți negative, acestea fiind prezentate în continuare.

Amazon Alexa este unul dintre cele mai cunoscute sisteme de automatizare și monitorizare a locuințelor, oferind compatibilitate cu sute de milioane de dispozitive periferice. Aplicația mobilă aferentă sistemului oferă un mod de utilizare fluent și intuitiv, fiind posibilă trimiterea de comenzi prin simpla comandă vocală emisă de utilizator. Ecosistemul Alexa permite conectare dispozitivelor de la o gamă largă de companii, printre care se numără Nest, Samsung și Philips. Dispozitivele compatibile cu sistemul pot fi găsite cu ușurință, având tag-ul de confirmare al compatibilității mereu încorporat².

Google Assistant este competitorul cel mai apropiat al sistemului menționat anterior. Aceasta prezintă o compatibilitate mai scăzută cu dispozitive de la alte companii în comparație cu Alexa, dar are o putere de interpretare a comenziilor utilizatorului mult mai mare. Acest aspect este datorat faptului că Google deține cel mai utilizat motor de căutare, dispunând astfel de algoritmi de căutare mai complecși. Conform ZDnet, Google Assistant are o acuratețe de interpretare a comenziilor de 92.9%, în timp ce Alexa de doar 79.8%. În ciuda compatibilității cu un număr mai redus de dispozitive periferice sistemul este compatibil cu majoritatea produselor oferite de companiile cunoscute precum Philips, Nest și Belkin³.

Wink Hub oferă utilizatorului posibilitatea de a alege orice dispozitive periferice, aspect care nu se regăsește în cadrul celor două sisteme menționate anterior. Caracterul de Plug and Play al sistemului combinat cu compatibilitatea cu mai multe dispozitive oferă utilizatorului libertate în construirea unui ecosistem complex cu un cost redus. Ca și forme de conectivitate locale se regăsesc Bluetooth LE și Wi-Fi. Sistemul vine și cu o aplicație mobilă dedicată care încorporează toate funcționalitățile standard⁴.

Samsung SmartThings Hub este o alternativă la sistemele de monitorizare și automatizare a locuințelor menționate anterior, oferind o gamă de dispozitive dedicate precum SmartThings Lightbulb și SmartThings Cam. Aplicația mobilă dedicată oferă toate funcționalitățile standardizate ale sistemelor Smart Home. Este de asemenea posibilă și integrarea Amazon Alexa sau Google Assistant în cadrul SmartThings Hub, aspect care nu este regăsit la cele trei sisteme menționate anterior⁵.

²<https://www.amazon.com/alexasmarthome/>

³<https://assistant.google.com/smart-home/>

⁴<https://www.wink.com/products/wink-hub/>

⁵<https://www.smartthings.com/>

3.3 Automatizarea imobilelor cu Arduino

Mediul de dezvoltare Arduino integrează toate funcționalitățile paradigmelor IoT, oferind suport pentru programarea diferitelor componente pentru realizarea comunicării bidirectionale dintre acestea. Astfel este ideal pentru realizarea proiectelor de automatizare și monitorizare a locuințelor, oferind dezvoltatorului posibilitatea de a adăuga mereu funcționalități suplimentare⁶.

Placa de dezvoltare Arduino Mega 2560 este bazată pe microcontrollerul ATMega2560 oferit de cei de la Atmel, fiind destinată programării cu ajutorul mediului de dezvoltare Arduino IDE. Astfel în urma conectării unui anumit număr de senzori și actuatoare se formează un sistem local de Smart Home complet, acesta având un mod de funcționare paralel. Datele sunt primite de la senzori la fiecare iterație a programului principal și sunt procesate corespunzător. Pentru depistarea anumitor parametri mai importanți se folosesc întreuperi, care notifică detectia fără întârzieri la orice moment al execuției. Acționarea dispozitivelor și a actuatoarelor locale se realizează cu ajutorul unui modul de relee, comandat în funcție de canalul dorit. Toate operațiile menționate anterior sunt executate de microcontrollerul ATMega2560.

Pentru realizarea comunicării în cadrul sistemului se folosesc interfețele seriale disponibile, acestea respectând standardul UART. Comunicarea cu exteriorul sistemului local se realizează cu ajutorul modulelor de wireless sau Bluetooth, avansul tehnic oferind module de dimensiune redusă cu consum de energie scăzut[6].

Întrucât partea locală a sistemului Smart Home necesită un consum mic de energie și o fiabilitate ridicată mediul de dezvoltare Arduino oferă suport pentru modulul de wireless dedicat ESP-8266. Acesta îndeplinește ambele cerințe menționate anterior, fiind ușor de programat și încorporat în hub-ul local. Datele de la senzori și actuatoare sunt prelucrate de microcontrollerul integrat în placă de dezvoltare Arduino Mega și sunt trimise cu ajutorul interfeței seriale la modulul esp. Modulul de wireless reprezintă calea de acces a sistemului către serviciul de Cloud, realizând cu acesta o comunicare bidirectională. În urma conectării cu succes la o rețea de internet modulul ESP-8266 rulează programul încărcat cu ajutorul mediului de dezvoltare Arduino IDE, așteptând primirea de date de pe partea interfeței seriale.[7]

Arduino este astfel ideal pentru realizarea sistemelor Smart Home de dimensiune redusă, oferind o fiabilitate crescută la un preț al componentelor scăzut. Mantenanța este de asemenea ușoară, căile de comunicare dintre module nefiind permanente. Prin integrarea interfețelor seriale, a întreupерilor și a numărului considerabil de pini placile de dezvoltare compatibile cu Arduino satisfac toate cerințele necesare implementării sistemelor de monitorizare și automatizare.

O alternativă la placile Arduino sunt cele oferite de Raspberry Pi, acestea având un consum de energie mai crescut și o complexitate mai ridicată. Plăcile Raspberry Pi sunt destinate sistemelor cu un consum de resurse mai ridicat⁷.

⁶<https://www.arduino.cc/en/software>

⁷<https://www.raspberrypi.org/>

Capitolul 4

Analiză și Fundamentare Teoretică

În cadrul acestui capitol se elaborează implementarea funcționalităților proiectului dintr-un punct de vedere teoretic, aspect necesar înțelegerei implementării practice prezentată pe parcursul capitolului următor. Sunt prezentate secțiuni precum scopul proiectului, cerințele sistemului și perspectiva tehnologică.

4.1 Scopul proiectului

Proiectul are ca scop implementarea unui sistem complet de automatizare și monitorizare al locuințelor în timp real și de la distanță, oferind o alternativă la produsele existente pe piața sistemelor Smart Home. În cadrul implementării se remarcă două părți accesibile utilizatorului, și anume partea locală a sistemului sau partea hardware și aplicația mobilă dedicată.

Partea locală oferă utilizatorului o monitorizare limitată din cadrul imobilului, acesta putând interacționa în mod direct cu aceasta. Conectarea de dispozitive externe la sistem este una ușoară, având un caracter Plug and Play.

Aplicația mobilă este intuitivă și ușor de utilizat, oferind utilizatorului toate funcționalitățile integrate în sistem. Aceasta oferă posibilitatea de monitorizare și trimitere de comenzi în timp real, de la distanță.

4.2 Cerințele sistemului

4.2.1 Cerințe funcționale

Identificarea cerințelor funcționale ale sistemului este elementară pentru înțelegerea procesului de implementare, oferind o perspectivă completă asupra modului de funcționare. Cerințele funcționale sunt preluate din cadrul aplicației mobile destinate monitorizării, aceasta oferind o versiune completă a funcționalităților sistemului în comparație cu partea

locală. Utilizarea sistemului implică existența unui singur tip de utilizator, conturile necesare logării în aplicație fiind asignate automat în funcție de sistemul local ales. În Tabelul 4.1 sunt prezentate toate cerințele funcționale din cadrul proiectului, alături de tipul de utilizator asignat.

Tabelul 4.1: Cerințele funcționale ale sistemului de monitorizare și automatizare

	Cerință funcțională	Rol
CF1	Autentificare	Utilizator
CF2	Control ială electrică	Utilizator
CF3	Monitorizare temperatură în timp real	Utilizator
CF4	Monitorizare umiditate în timp real	Utilizator
CF5	Programare termostat	Utilizator
CF6	Control termostat	Utilizator
CF7	Monitorizare stare instalație de climatizare în timp real	Utilizator
CF8	Monitorizare stare instalație de încălzire în timp real	Utilizator
CF9	Control instalație de climatizare	Utilizator
CF10	Control instalație de încălzire	Utilizator
CF11	Monitorizare stare instalație de irigare în timp real	Utilizator
CF12	Programare instalație de irigare	Utilizator
CF13	Control instalație de irigare	Utilizator
CF14	Monitorizare stare sistem securitate în timp real	Utilizator
CF15	Armare sistem de securitate	Utilizator
CF16	Dezarmare sistem de securitate	Utilizator
CF17	Confirmare alertă incendiu	Utilizator
CF18	Confirmare alertă surgeri gaz metan	Utilizator
CF19	Confirmare alertă detectie mișcare	Utilizator

4.2.2 Cerințe non-funcționale

Cerințele non-funcționale sunt esențiale pentru precizarea standardelor la care trebuie să se ridice implementarea proiectului, arătând modul în care se vor realiza cerințele funcționale menționate anterior. În continuare sunt elaborate cerințele non-funcționale regăsite în cadrul proiectului de automatizare și monitorizare a locuințelor.

Durabilitatea

Durabilitatea reprezintă abilitatea unui produs fizic de a funcționa pe o durată îndelungată de timp, fără a fi nevoie de mentenanță periodică sau alte intervenții din exterior. Aceasta este un element foarte important pe piată ecosistemelor Smart Home, utilizatorii căutând sisteme cu o durată extinsă de funcționare.

În cadrul sistemului dezvoltat partea locală satisfac această cerință, componentele necesare implementării fizice fiind alese astfel încât să respecte principiul unei perioade de utilizare extinse. De asemenea menenanța este una ușoară, toate componentele fiind Plug and Play, fără conexiuni permanente care ar putea implica dificultăți la înlocuire.

Extensibilitatea

Extensibilitatea este principiul care presupune posibilitatea de îmbunătățire a unui sistem informatic prin adăugarea de funcționalități noi, păstrând sau îmbunătățind cele deja existente. Procesul de extindere nu trebuie să implice un grad ridicat de dificultate, sistemul fiind construit pentru a putea fi îmbunătățit în viitor.

Sistemul de automatizare și monitorizare al imobilelor implementat în cadrul acestui proiect respectă principiul extensibilității, fiind structurat pe trei nivele diferite pentru adăugarea cu ușurință de componente/funcționalități noi. Astfel fiecare funcție nouă se implementează separat în cadrul părții locale, în cadrul serviciului de Cloud și în cadrul aplicației mobile dedicate. Unificarea se realizează cu ușurință, respectând principiile de implementare utilizate anterior.

Toleranța la defecte

Toleranța la defect reprezintă proprietatea unui sistem de a putea fi utilizat în continuare în ciuda funcționării incorecte a unor componente. Este o cerință non-funcțională mandatorie proiectelor realizate conform standardelor pieții, fiind un factor decisiv pentru mulți posibili clienți.

Sistemul actual prezintă o toleranță la defecte ridicată, defectarea unei componente din cadrul părții hardware neavând efecte asupra monitorizării celorlalți parametrii. De asemenea sistemul are o toleranță ridicată și la posibilele erori software care pot apărea pe parcurs, precum transmiterea datelor într-un format greșit, ambele părți care comunică având un proces de verificare conform unui format prestabilit. Posibila deconectare de la rețeaua de internet a părții locale sau a aplicației mobile nu prezintă de asemenea riscuri, fiind stocate ultimele date primite în cadrul serviciului de Cloud, urmând să fie astfel procesate după reconectare.

Integrabilitatea

Integrarea sistemelor reprezintă posibilitatea interconectării unui număr mare de sub-sisteme cu scopul de a obține un singur sistem, modulele comunicând între ele simultan. Astfel se poate realiza conexiunea dintre partea software și cea hardware cu succes cu ajutorul unor protocoale specifice.

În cadrul ecosistemului Smart Home implementat în proiect integrabilitatea este regăsită la conectarea părții locale/hardware cu aplicația mobilă, aspect realizat cu ajutorul unui serviciu de Cloud destinat transmiterii de date bidirectionale. Astfel se obține un

sistem de monitorizare complet, cele două componente comunicând în permanentă între ele.

Mentenanța

Mentenanța reprezintă nivelul de dificultate aferent întreținerii unui sistem informatic, prin corectarea anumitor posibile erori sau defecțiuni. Un nivel scăzut de dificultate al mentenanței reprezintă un criteriu important în alegerea unui ecosistem de automatizare și monitorizare al domiciliului, fiind un aspect la fel de important precum durabilitatea.

După cum este menționat și mai sus, sistemul are o structură astfel proiectată încât schimbarea componentelor integrate sau a conexiunii dintre acestea să se realizeze cu ușurință, fără a fi nevoie de instrumente sau tehnologii speciale. De asemenea posibilele erori software apărute pe partea locală se pot realiza prin simpla resetare a componentei hardware sau prin deconectarea sursei de alimentare.

Performanța

Performanța reprezintă randamentul de funcționare al unui sistem informatic, fiind un criteriu important în realizarea oricărui ecosistem digital. Această cerință non-funcțională poate fi cuantificată după diverse criterii cum ar fi latență, durata execuției unor task-uri sau calitatea execuției acestora.

Sistemul de automatizare și monitorizare implementat în cadrul proiectului de față folosește câte un canal pentru fiecare direcție de comunicare, datele fiind trimise simultan cu ajutorul codificării locale. Acest aspect conferă o întârziere de transmitere a datelor de aproximativ o secundă, un delay neglijabil având în vedere posibilitatea de utilizare de la distanță. Realizarea funcțiilor locale este imediată în momentul decodificării comenzielor primite, partea locală având un mod de funcționare paralel.

Securitatea

Securitatea reprezintă cerință non-funcțională prin care un sistem trebuie să respecte anumite standarde de protejare a datelor stocate, respectiv de asigurare a păstrării datelor confidențiale ale utilizatorilor.

Proiectul actual folosește pentru serviciul de Cloud platforma Firebase oferită de cei de la Google, fapt care îi conferă un nivel de securitate ridicat, atât pe partea stocării și transmiterii datelor cât și pe partea de autentificare a utilizatorilor. Astfel posibilitatea de pierderi de date este scăzută, având un randament mai mare decât în cadrul implementării transmisiei de date cu requesturi HTTP nesecurizate.

Utilizabilitatea

Utilizabilitatea reprezintă nivel de dificultate pe care îl implică utilizarea unui sistem informatic, dar și calitatea generală a experienței de utilizare a acestuia. Este un criteriu

important pentru majoritatea utilizatorilor în momentul unei achiziții, fiecare client dorind un modul hardware/software ușor de utilizat, care să ofere o interfață fluidă.

Sistemul Smart Home implementat în cadrul proiectului oferă o parte hardware cu toate componentele conectate cu ușurință, având un caracter Plug and Play. Meniu local are de asemenea o interfață navigabilă cu ușurință, oferind un set de funcționalități esențiale.

Aplicația mobilă are un design plăcut și atractiv, fiecare element din interfață grafică fiind amplasat în ordinea priorității. Reprezentarea datelor monitorizate este una clară și sugestivă, trimiterea de comenzi fiind facilitată de asemenea prin grafici specifice. Nu există delay local la interacțiunea cu interfața grafică, acesta apărând doar la transmiterea datelor.

4.3 Perspectiva tehnologică

4.3.1 Interfețe de comunicare serială

Comunicarea UART

Comunicarea UART este reprezentată de o interfață serială care funcționează în mod asincron, fără existența unui semnal de ceas sau Clock. Prințipiu pe care este bazat acest tip de comunicare este existența unui transmitter numit și Tx și un receiver numit și Rx. Interconectarea a două dispozitive pentru transmiterea de date de la primul la cel de al doilea este prezentată în Figura 4.1. Datele sunt primite într-un mod paralel înaintea transmiterii, urmând să fie convertite și transmise sub forma unui flux de biți. Formatul fluxului transmis este un bit de start, între 5 și 9 biți de date, între 0 și 1 biți de paritate și între 1 și 2 biți de stop. Este cel mai simplist tip de comunicare asincronă, fiind utilizată pe scară largă în cadrul plăcilor de dezvoltare precum Arduino.

Folosirea comunicației UART în cadrul sistemului local al proiectului este necesară în momentul transmiterii de date dintre diferite module, dar și în cazul depănării sau programării unui modul individual.

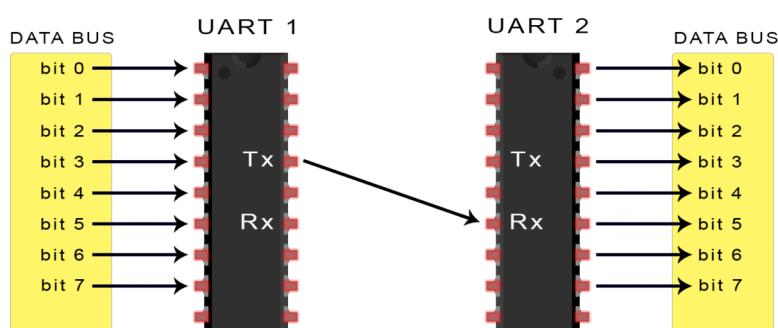


Figura 4.1: Conectarea interfeței UART¹

Comunicarea SPI

Comunicarea SPI este un tip de comunicare serială de tipul master-slave, unde un device de tipul master poate controla un număr mai mare de dispozitive de tipul slave. Avem astfel o interfață serială sincronă, cu ceas sau Clock, care necesită 4 conexiuni pentru a realiza procesul de comunicare. Cele patru semnale care necesită conectare sunt SCLK, SS, MOȘI și MISO. Semnalul SCLK este necesar sincronizării transmiterii datelor, fiind semnalul de Clock. Semnalul SS este semnalul de selecție al slave-ului, selectând astfel destinația transmiterii datelor. Pinul MOȘI este utilizat pentru transmiterea datelor de la master la slave, în timp ce MISO este utilizat pentru trimiterea datelor de la slave la master. În Figura 4.2 este prezentat modul de funcționare al interfeței seriale SPI. Acest tip de comunicare serială este foarte des întâlnită la modulele de citire carduri RFID.

Folosirea comunicației SPI în cadrul sistemului local al proiectului este utilă pentru anumite componente specifice, crescând astfel posibilitatea de extindere a funcționalităților.

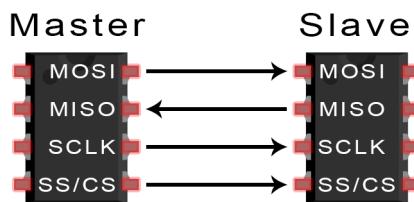


Figura 4.2: Conectarea interfeței SPI²

Comunicarea I2C

Comunicarea I2C reprezintă un tip de comunicare sincronă, cu semnal de ceas sau Clock, care are un mod de transmitere a datelor serial. Este de asemenea o comunicare de tipul master-slave, însă că avantajul apare posibilitatea de a avea un număr mai mare de device-uri master care controlează un număr ridicat de slave-uri. Există astfel doar doi pini pentru realizarea comunicării, SDA și SCL. SDA este utilizată pentru transmiterea și primirea de date dintre master și slave, iar SCL este folosit pentru transmiterea semnalului de Clock. Formatul unei instrucțiuni trimise cu ajutorul liniei SDA este primii 7 sau 10 biți pentru adresa cadrului, bitul de citire/scriere, bit de acknowledge, 8 biți pentru primul cadru, bit de acknowledge, 8 biți pentru al doilea cadru de date, bit de acknowledge. Transmiterea începe când pe linia SDA se identifică un front descrescător de ceas înaintea liniei SCL și se încheie în momentul în care pe linia SDA se identifică un front crescător de ceas înaintea liniei SCL. În Figura 4.3 este prezentat modul de conectare al interfeței I2C. Acest tip de comunicare este utilizată în cazul dispozitivelor cu multe conexiuni, cum ar fi conectarea unui display LCD la un microcontroller.

¹<https://www.circuitbasics.com/basics-uart-communication/>

²<https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>

³<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>

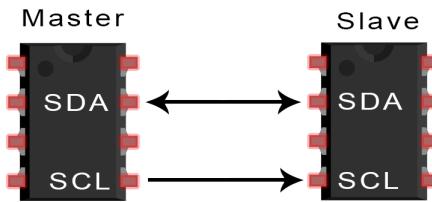


Figura 4.3: Conectarea interfeței I2C³

Precum în cazul interfetei SPI, comunicarea I2C este utilă în cadrul proiectului pentru anumite componente specifice care au nevoie de un număr mare de conexiuni pentru a putea fi controlate, crescând astfel posibilitatea de extindere a funcționalităților.

4.3.2 Componente hardware utilizate

Arduino Mega

Placă de dezvoltare Arduino Mega 2560 este bazată pe microcontrollerul ATmega2560 oferit de cei de la Atmel, oferind o performanță crescută cu un consum de resurse scăzut. Placa dispune de 54 pini digitali, 16 intrări analogice și 4 porturi seriale. Dintre cei 54 pini digitali 15 se pot utiliza ca și ieșiri PWM. Programarea microcontrollerului se realizează cu ușurință cu ajutorul magistralei USB integrate, fiind disponibil un spațiu de stocare intern de 256 KB pentru încărcarea programelor. Voltajul de operare al plăcii de dezvoltare este de 5V, oferind și o alimentare de 3.3V pentru conectarea modulelor specifice. Arduino Mega 2560 încorporează și interfețe seriale precum UART, I2C și SPI, utile în momentul conectării de dispozitive noi⁴.

Display LCD

Modulul lcd permite afișarea unui număr de 32 de caractere, împărțite pe două rânduri de afișaj. Acesta dispune de 16 pini de operare, aspect care poate fi redus la doar 4 pini prin folosirea interfeței I2C. Pe lângă pinii de alimentare modulul dispune de un pin de contrast și doi pini de reglare a intensității luminoase, 8 pini de date și încă 3 pini specifici, aceștia fiind RS,RW și EN. Pinul RS este utilizat pentru selectarea registrului de comandă, RW este utilizat pentru citire/scriere și EN este utilizat pentru activarea anumitor operații. Modulul poate fi interfațat paralel prin folosirea pinilor menționați anterior sau în mod serial prin adăugarea unui modul de I2C⁵.

Este ideal în cadrul proiectelor hardware unde este necesară afișarea în timp real a unor parametri sau interacțiunea directă cu sistemul.

⁴<https://microcontrollerslab.com/introduction-arduino-mega-2560/>

⁵<https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>

Tastatura matriceala

În cadrul tastaturii matriceale de 3 coloane și 4 rânduri sunt disponibile 12 caractere. În Figura 4.4 este prezentat modul de funcționare al acesteia, selectarea unui caracter realizându-se prin combinarea rândului și a coloanei pe care se regăsește. Operația de selecție se realizează cu ajutorul unor butoane tactile aflate la intersecția rândurilor cu coloanele. Astfel modulul are nevoie de un număr de 7 semnale, și anume 3 semnale pentru selectarea coloanei și 4 semnale pentru selectarea rândului. Tastatura nu depinde de semnalul de ceas, având un mod de funcționare asincron și simplist.

Este un modul ideal sistemelor hardware care au nevoie de input direct de la utilizator.

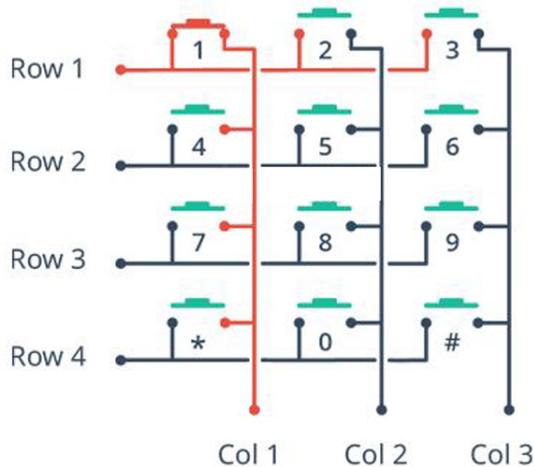


Figura 4.4: Modul de funcționare al tastaturii matriceale⁶

Modul de relee

Modulul de relee cu patru canale este alcătuit din patru relee alimentate de aceeași sursă de tensiune și controlate de semnale diferite. Astfel modulul dispune de 6 semnale de intrare, și anume Vcc, Gnd, IN1, IN2, IN3 și IN4. Ultimele patru semnale sunt utilizate pentru activarea și respectiv dezactivarea fiecărui releu în parte, în timp ce Vcc și Gnd sunt folosite pentru alimentarea întregului modul. Structura internă a unui releu cu semnalele aferente este descrisă în Figura 4.5.

Fiecare releu are trei semnale de intrare și alte trei semnale de control. Printre semnalele de intrare se remarcă Vcc și Gnd destinate alimentării, alături de semnalul Input care controlează starea releului. Ca și semnale de control sunt identificate semnalele

⁶<https://lastminuteengineers.com/arduino-keypad-tutorial/>

de Normal deschis/NO, Contact comun/COM și Normal închis/NC. Semnalul Normal deschis este utilizat pentru controlul circuitelor deschise care trec în starea de închis în urma activării releului, iar semnalul Normal închis este utilizat pentru circuitele închise care trec în starea de deschis că și urmare a acționării releului. Semnalul Contact comun este folosit la ambele tipuri de circuite, după cum este precizat și în denumirea acestuia.

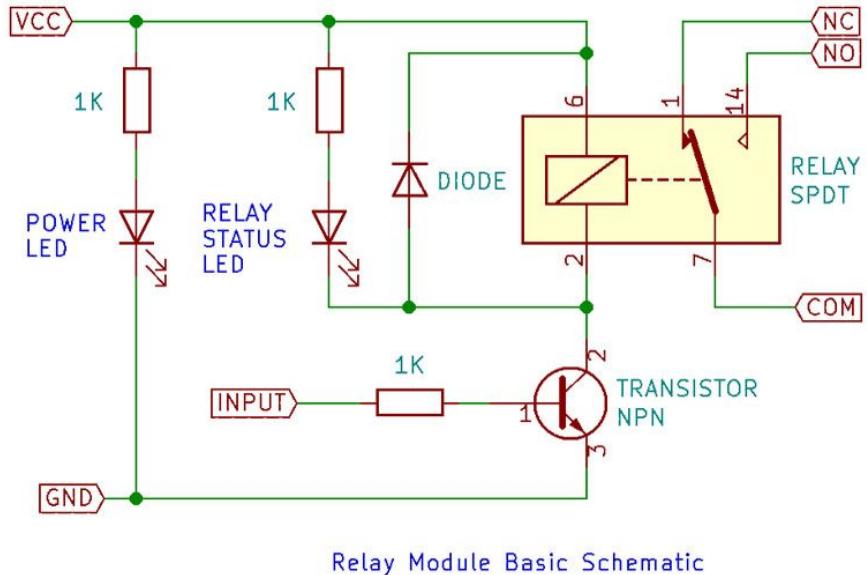


Figura 4.5: Structura internă a unui releu⁷

Cititor RFID

Modulul MFRC522 este un cititor RFID dedicat, utilizând tehnologia de identificare a frecvențelor radio. Astfel se utilizează câmpuri electromagnetice pentru transferul de date între un cititor și un chip aflat pe partea destinației. Transmiterea datelor se realizează fără atingere, de la o distanță redusă, prin executarea acțiunii de apropiere a transponder-ului RFID de cititor. Frecvența câmpului electromagnetic la care modulul lucrează pentru citirea cardurilor și a cartelelor specifice este 13.56MHz, respectând standardul ISO 14443A. O reprezentare a modului de funcționare al tehnologiei RFID este prezentată în Figura 4.6.

Modulul curent comunică cu microcontrollerul ATmega2560 cu ajutorul interfeței seriale SPI, disponând de asemenea de pini aferenți protoalelor UART și I2C. Pinii modulului pentru comunicarea SPI sunt SS, SCK, MOȘI, MISO, IRQ, GND, RST și Vcc. După cum a fost menționat și în secțiunea interfețelor de comunicare serială pinul SS are rolul de a selecta dispozitivul slave, pinul SCK are rolul de a sincroniza transmisia prin

⁷<https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet>

propagarea semnalului de ceas, pinul MO_SI este destinat transmiterii de date de la master la slave, pinul MISO este destinat transmiterii de date de la slave la master, pinul IRQ este destinat întreruperilor, pinul RST este destinat resetării modulului iar GND și Vcc sunt utilizate pentru alimentarea acestuia. Interfețele UART și I2C pot fi folosite de asemenea la transmiterea datelor, fiind însă nevoie de modificări și adaptări hardware ulterioare.

Modulul de cititor RFID este util în cadrul sistemelor care prezintă încorporarea interacțiunii dintre utilizator și sistem prin scanarea unor componente, precum carduri sau cartele. Are de asemenea ca utilizare implementarea în cadrul sistemelor de securitate, oferind risc de pierdere a datelor redus datorită distanței mici de funcționare.

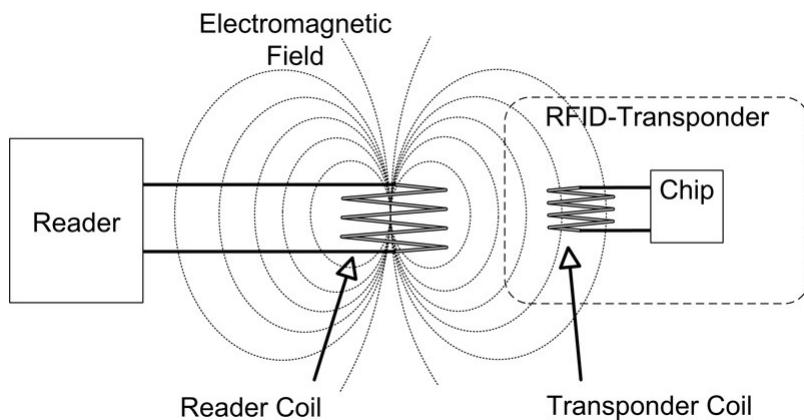


Figura 4.6: Modul de funcționare al tehnologiei RFID⁸

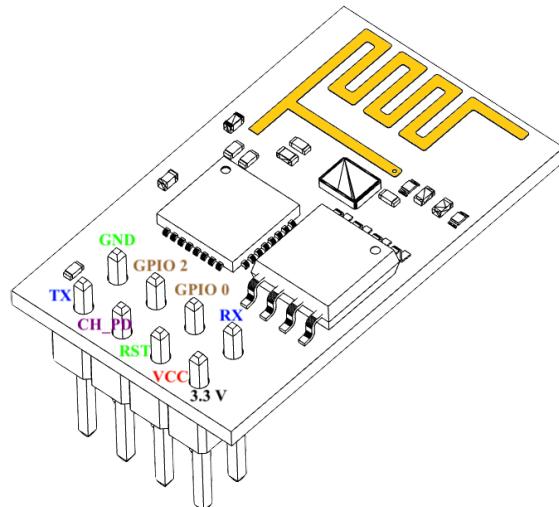
Modul Wireless

Modulul Wireless ESP-8266 este utilizat pentru conectarea sistemelor hardware la internet, oferind o alternativă ieftină și fiabilă. Acesta oferă un microcontroller integrat, fapt care facilitează scrierea programelor în cadrul memorie de 80KB oferită de modul. Astfel ESP-8266 are un mod de funcționare autonom, nefind necesară conectarea la un microcontroller extern. ESP-8266 integrează de asemenea protocoalele TCP/IP, fapt care îi conferă o compatibilitate ridicată cu serviciile de internet actuale. Structura modulului formată din pini, microcontroller și antenă este prezentată în Figura 4.7. Modulul are de asemenea un consum de energie scăzut, putând fi astfel integrat în cadrul sistemelor cu o capacitate de alimentării scăzută. Aceasta funcționează la o tensiune de 3.3V, cu o toleranță de până la 3.6V.

Modulul comunică cu microcontrolere externe cu ajutorul interfeței seriale UART, oferind un număr de 8 pini integrați. Aceștia sunt GPIO-0, GPIO-2, RXD, TXD, CHPD,

⁸https://www.researchgate.net/The-inductive-communication-between-reader-and-a-tag-using-coils_fig6_283498836

⁹<https://en.wikipedia.org/wiki/ESP8266>

Figura 4.7: Structura modulului ESP-8266⁹

RST, VCC și GND. Pentru transmiterea și primirea de date de la alte module se folosesc semnalele RXD și TXD, conectate conform modului de utilizare al interfeței UART. Modul de realizare al legăturilor dintre ESP-8266 și modulele externe se realizează în mod diferit în funcție de cazul de utilizare¹⁰. Astfel pentru scrierea programului din memoria modulului Wireless se realizează următoarele legături cu o anumită placă de dezvoltare compatibilă:

- GND,GPIO-0 > GND
- GPIO-2,RST > neconectat
- RXD > RX
- TXD > TX
- CHPD,VCC > 3.3V

Pentru utilizarea modulului wireless în paralel cu alt microcontroller care va primi, prelucra și trimite înapoi date se realizează următoarele legături:

- GND > GND
- GPIO-2,GPIO-0,RST > neconectat
- RXD > TX
- TXD > RX

¹⁰<https://create.arduino.cc/projecthub/pratikdesai/how-to-program-esp8266-esp-01-module-with-arduino-uno-598166>

- CHPD,VCC > 3.3V

Modulul ESP-8266 este util în cadrul sistemelor locale hardware pentru conectarea acestora la internet, oferind un mod de funcționare stabil, cu un consum de energie redus. Există diverse protocole și librării destinate acestui modul, fiind compatibil cu majoritatea serviciilor de Cloud existente pe piață în momentul actual.

Buzzer

Buzzerul este un dispozitiv electric responsabil de reproducerea sunetelor, fiind cunoscut și ca difuzor Piezo. Acest modul are un consum mic de energie, operând la o gama largă de voltaje, între 3 și 250V. Acesta are un nivel ridicat de presiune sonoră în comparație cu alternative precum difuzorul magnetic. Fiecare buzzer piezo conține o componentă piezoelectrică, aceasta fiind formată din ceramică piezoelectrică și metal. Componenta este responsabilă pentru producerea sunetelor prin intermediul vibrației, fenomen care are loc în urma aplicării unui voltaj alternativ.

Modulul are 3 pini, și anume VCC,GND și I/O. Pinii VCC și GND sunt utilizati pentru alimentarea buzzerului cu un voltaj între 3 și 250V, pinul I/O fiind responsabil pentru controlul frecvenței sunetului. Difuzoarele piezo care oferă un al treilea pin (I/O) se numesc și buzzere piezo cu feedback. Feedback-ul controlează nivelul de vibrație al componentei piezoelectrice, astfel creând sunete cu frecvențe diferite.

Buzzerele piezo sunt utilizate în proiectele care necesită sunete de o calitate scăzută dar cu intensitate ridicată, precum alarme sau sunete ale interacțiunii cu interfața sistemului. Acestea vin încapsulate sub diverse forme, componenta generală fiind aceeași în toate cazurile.

Senzor detectie flăcări

Pentru detectia flăcărilor se folosește un modul bazat pe un senzor infraroșu sau IR, cu detectie pe o distanță scurtă. Tehnologia infraroșu este utilizată în mare parte în domeniul sistemelor termice, folosind o radiație electrică cunoscută și că IR care are o lungime de undă mai mare decât cea a luminii vizibile.

Modulul destinat detectiei flăcărilor are o distanță maximă de identificare de un metru, având o sensibilitate ridicată la radiațiile IR cuprinse în intervalul 760-1100 nm. În Figura 4.8 este prezentată rază spectrală a radiației IR în comparație cu celelalte tipuri de radiații. Astfel modulul folosește radiația de tipul NIR pentru procesul de detectie.

Modulul dispune de patru pini de conectare, aceștia fiind VCC,GND,AO și DO. Pinii VCC și GND sunt utilizati pentru alimentarea la 5V a modulului, în timp ce pinii AO și DO sunt utilizati pentru transmiterea datelor de la senzor. Pinul AO transmite valoarea analogică rezultată în urma citirii senzorului, în timp ce DO devine activ în momentul

¹¹https://www.researchgate.net/figure/Spectral-range-for-near-infrared-NIR-and-mid-infrared-MIRShowing-wavelengths-nm-fig1_338633707

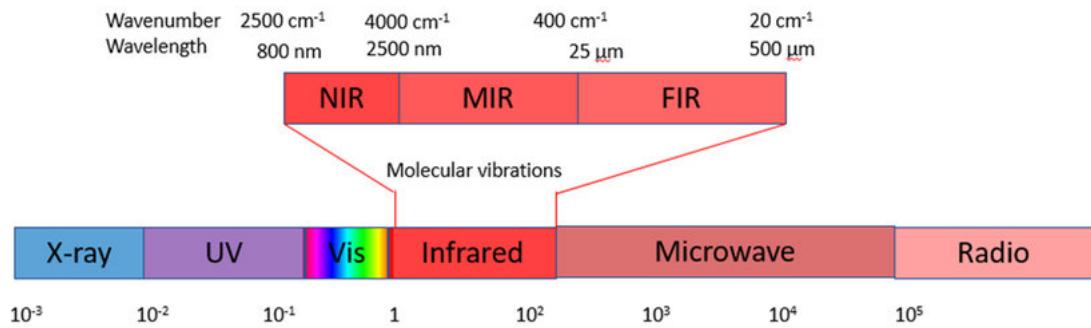


Figura 4.8: Radiația infraroșie în cadrul spectrului electromagnetic¹¹

detectiei flăcărilor. Senzitivitatea pragului de detectie pentru pinul DO este reglată cu ajutorul unui potențiometru integrat în modul¹².

Aceste module sunt utilizate în sistemele unde este necesară detectia flăcărilor la o distanță scurtă de senzor, fiind declanșate alerte sau acțiuni în urma acesteia.

Senzor detectie gaz

Modulul de detectie MQ-4 este destinat depistării gazului metan și a gazului natural comprimat. Acesta are o sensibilitate mică la detectia elementelor precum alcool și fum, oferind un răspuns scurt la depistarea gazelor și o fiabilitate ridicată. Modulul necesită o perioadă de pre-încălzire pentru realizarea procesului de detectie cu succes, fapt datorat multitudinii de materiale din care este alcătuit. Structura și materialele care intră în componența senzorului sunt prezentate în Figura 4.9.

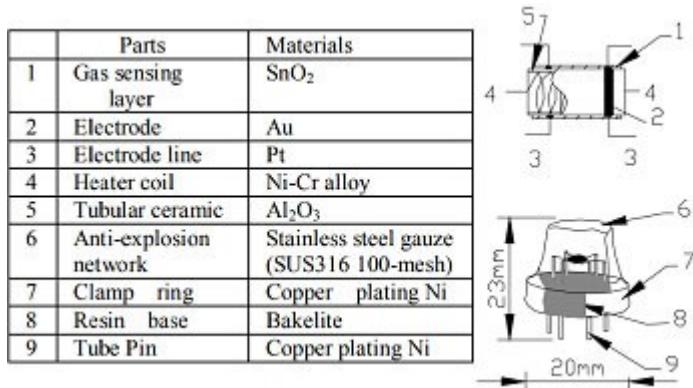


Figura 4.9: Structura internă a senzorului MQ-4¹³

¹²<https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing>

¹³https://linksprite.com/wiki/index.php?title=MQ-4_Semiconductor_Sensor_for_Natural_Gas

Pinii disponibili în cadrul modulului MQ-4 sunt VCC,GND,AO și DO. Pentru alimentarea modulului la 5V se folosesc pinii VCC și GND, în timp ce transmiterea datelor se realizează cu ajutorul pinilor AO și DO. AO reprezintă valoarea analogică rezultată în urma citirii senzorului de detectie, în timp ce DO este ieșirea care devine activă în momentul detectiei de gaz. Pragul de detectie aferent pinului DO se reglează cu ajutorul potențiometrului integrat în modul.

Modulele MQ-4 sunt utile în cadrul sistemelor care impun detectia gazelor, oferind posibilitatea de depistare timpurie a eventualelor scurgeri care pot implica un nivel de pericol ridicat.

Senzor detectie mișcare

Senzorul de detectie al mișcării, numit și senzor PIR sau Passive Infra Red Sensor, este un modul care detectează mișcarea cu ajutorul radiației infraroșii. Corpul uman emite un anumit nivel de IR, prin schimbarea poziției într-un cadru bine determinat declanșând un rezultat pozitiv al detectiei. În exteriorul suprafeței senzorului există o lentilă Fresnel necesară extinderii spectrului de detectie pe toate direcțiile de deplasare. În Figura 4.10 este explicat modul de funcționare al detectiei bazate pe IR, elementul termic emis de corpul uman fiind crucial. Senzorii de acest tip sunt o variantă de identificare a mișcării ieftină, fiabilă și fără consum ridicat de energie.

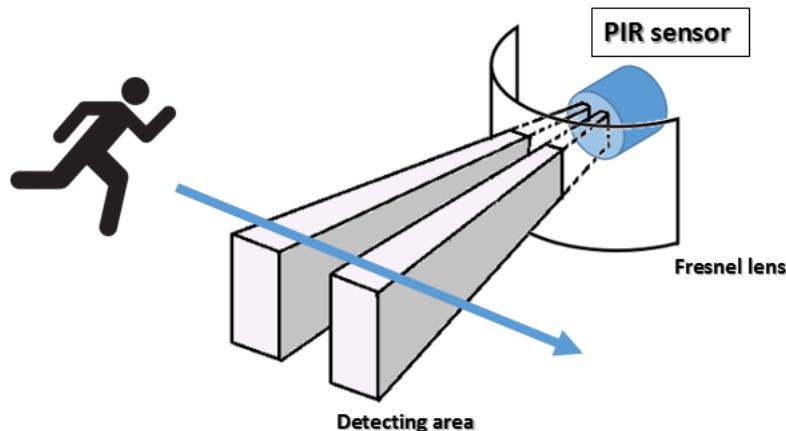


Figura 4.10: Modul de detectie al mișcării de către senzorul PIR¹⁴

Modulul PIR oferă un număr de trei pini integrați, aceștia fiind VCC,GND și Out. Alimentarea la 5V se realizează cu ajutorul pinilor VCC și GND, pinul out devenind activ

¹⁴<https://medium.com/@ariesiitr/home-automation-and-pir-passive-infrared-sensor-5f9592863b20>

în momentul detecției mișcării. Reglarea senzitivității se face în mod automat în momentul alimentării, durând aproximativ 60 de secunde.

Senzorii de detecție ai mișcării bazați pe radiații IR sunt utili în cadrul sistemelor de monitorizare, oferind o alternativă ieftină și fiabilă. Sistemele de acest tip pot fi integrate și în cadrul ecosistemelor Smart Home, oferind utilizatorilor o experiență de monitorizare extinsă.

Senzor de temperatură și umiditate

Modulul DHT11 este utilizat pentru monitorizarea în timp real a temperaturii și a umidității, oferind un cost redus și o fiabilitate crescută. Acesta folosește un senzor de umiditate capacitive și un termistor pentru măsurarea celor doi parametri. Termistorul este un rezistor a cărui rezistență depinde de temperatura externă, fiind astfel ideală pentru măsurarea acestui parametru. Structura senzorului capacitive destinaț măsurării umidității este prezentată în Figura 4.11. Aceasta utilizează un electrod superior și unul inferior, despărțiti de un material pentru menținerea umidității colectate. Acuratețea este de ± 2 grade Celsius pentru monitorizare temperaturii și 5% pentru monitorizarea umidității.

Modulul oferă trei pini, și anume VCC, GND și Signal. Pinii VCC și GND sunt utilizați pentru alimentarea modulului DHT11 la 3.3V sau la 5V. Pinul Signal este utilizat pentru transmiterea simultană a temperaturii și a umidității. Decodificarea celor doi parametrii se face în cadrul componentei conectate la modul în urma primirii a 40 biți de date.

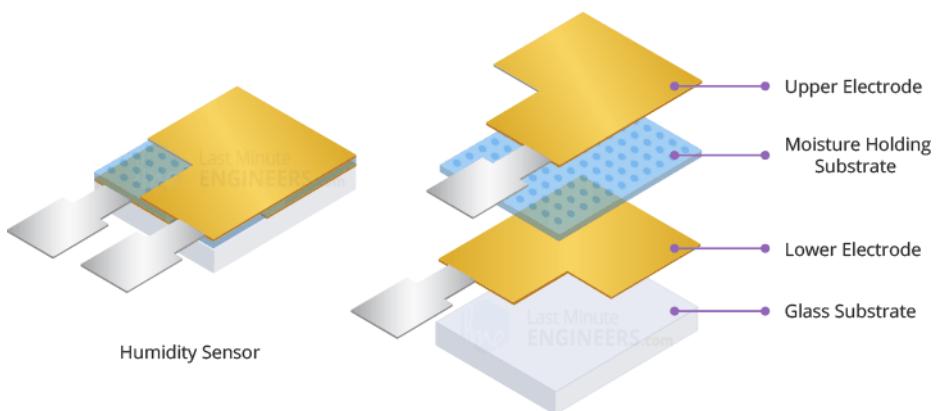


Figura 4.11: Structura unui senzor de umiditate capacitive¹⁵

Modulele de acest tip sunt utilizate în cadrul sistemelor care necesită monitorizarea în timp real a parametrilor din imobil/zona de activitate, spre exemplu în cadrul ecosistemelor Smart Home.

¹⁵<https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/>

4.3.3 Tehnologii utilizate

Firebase Realtime Database

Firebse Realtime Database este o bază de date de tipul NoSQL bazată pe serviciul Firebase oferit de cei de la Google. Este o baza de date care funcționează cu ajutorul Cloud-ului, oferind clientilor sincronizare în timp real. Ca și caracteristică specifică se remarcă faptul că baza de date rămâne activă și după deconectarea din cadrul aplicației mobile care o utilizează, orice schimbare de la un alt client fiind astfel înregistrată. Este ideală pentru realizarea comunicării dintre două părți ale unui sistem informatic, oferind o viteză de acces și de scriere rapidă. Pe lângă canalele de date care sunt actualizate în timp real în cadrul Firebase Realtime Database există și posibilitatea de inserare de instanțe a datelor, fiind posibilă generarea de snapshot-uri pentru analiză și lucrul cu date din trecut¹⁶.

Acest serviciu de Cloud este ideal pentru realizarea comunicării în cadrul sistemului Smart Home deoarece oferă o viteză de transfer ridicată, stabilitate și un nivel crescut al securității. Fiind un serviciu oferit de cei de la Google actualizările sunt frecvente, aducând mereu îmbunătățiri la capitolul latentă și securitatea datelor.

Flutter

Flutter este un framework destinat design-ului de aplicații mobile fluide și intuitive, fiind bazat pe limbajul de programare Dart. Este un framework publicat de cei de la Google, fapt care ii conferă o compatibilitate integrată cu serviciul de Cloud menționat anterior, și anume Firebase. Că și caracteristică specifică se remarcă folosirea Null safety-ului, aspect care oferă aplicațiilor realizate cu ajutorul acestui framework stabilitate pe parcursul rulării, fără excepții aruncate de un parametru null care să afecteze experiența de utilizare. Flutter este destinat construirii de aplicații pentru dispozitivele mobile care rulează sistemele Android și IOS. Widget-urile predefinite conferă acestui mediu de dezvoltare posibilitatea de a crea interfețe atractive cu ușurință, partea de backend fiind ușor încorporată în spatele acestora¹⁷.

Framework-ul prezentat anterior este ideal pentru realizarea unei aplicații de monitorizare, oferind conexiunea integrată cu serviciul de Cloud Google Firebase și widget-uri dedicate pentru vizualizarea de date în timp real și trimiterea de comenzi.

Gradle

Gradle este un tool folosit pentru realizarea build-ului proiectelor, oferind flexibilitate pentru construirea oricărui tip de software. Ca și caracteristici speciale se remarcă nivelul crescut al performanței și al eficienței prin rularea exclusivă a task-urilor a căror intrare sau ieșire a suferit o schimbare. Un alt avantaj este reprezentat de integrarea

¹⁶<https://firebase.google.com/>

¹⁷<https://flutter.dev/>

acestui tool în cadrul IDE-ului Android Studio și compatibilitatea acestuia cu framework-ul Flutter¹⁸.

Tool-ul prezentat anterior este util în momentul dezvoltării aplicațiilor mobile, fiind ideal pentru adăugarea de funcționalități noi. Astfel prin implementarea de secțiuni noi în cadrul aplicației bazate pe Dart și dezvoltate cu Flutter se evaluează doar noua parte în cadrul procesului de build, timpul de execuție al acestei etape fiind astfel mult mai scurt.

¹⁸<https://gradle.org/>

Capitolul 5

Proiectare de Detaliu și Implementare

În cadrul acestui capitol se elaborează implementarea funcționalităților proiectului astfel încât dezvoltarea și întreținerea ulterioară a sistemului să fie posibilă. Sunt prezentate secțiuni precum arhitectura sistemului, descrierea diferitelor module și diagrame.

5.1 Arhitectura sistemului

Sistemul elaborat în cadrul proiectului este structurat pe trei nivele, și anume partea locală, partea de Cloud și partea aplicației mobile.

Partea locală a sistemului este controlată de placă de dezvoltare Arduino Mega 2560, bazată pe microcontrollerul ATmega 2560 oferit de compania Atmel. Aceasta a fost programată cu ajutorul mediului de dezvoltare Arduino, fiind compatibilă cu toate modulele conectate. Sunt disponibili senzori care oferă informații necesare monitorizării în timp real și un modul de relee pentru controlul dispozitivelor/actuatorilor din cadrul imobilului. Pentru realizarea conexiunii la internet a sistemului local și implicit comunicarea cu partea de Cloud este utilizat modulul Wireless ESP 8266. Acesta comunică în mod bidirectional cu placa de dezvoltare Arduino Mega cu ajutorul interfeței seriale integrate, procesând datele primite sau necesare trimiterii după un anumit format specific.

Partea de Cloud a sistemului este hostată cu ajutorul serviciului Google Firebase, fiind folosită o baza de date de tipul Real-time pentru transmiterea datelor de la partea locală la aplicația mobilă și în sens invers. Sunt disponibile două canale de comunicare diferite astfel încât latența de comunicare să fie minimă.

Partea aplicației mobile este direct disponibilă utilizatorului în urma autentificării într-un cont valid, toate funcționalitățile sistemului fiind prezente în interfața acesteia. Informațiile sunt primite în timp real de la serviciul Firebase Realtime Database, trimiterea de comenzi realizându-se de asemenea cu un delay nesemnificativ datorită celui de al doilea canal de date disponibil în Cloud.

Arhitectura sistemului este prezentată sub forma unei scheme bloc în Figura 5.1, cele trei nivele regăsite în cadrul implementării fiind delimitate cu ajutorul liniilor punctate.

Mobile App System Side

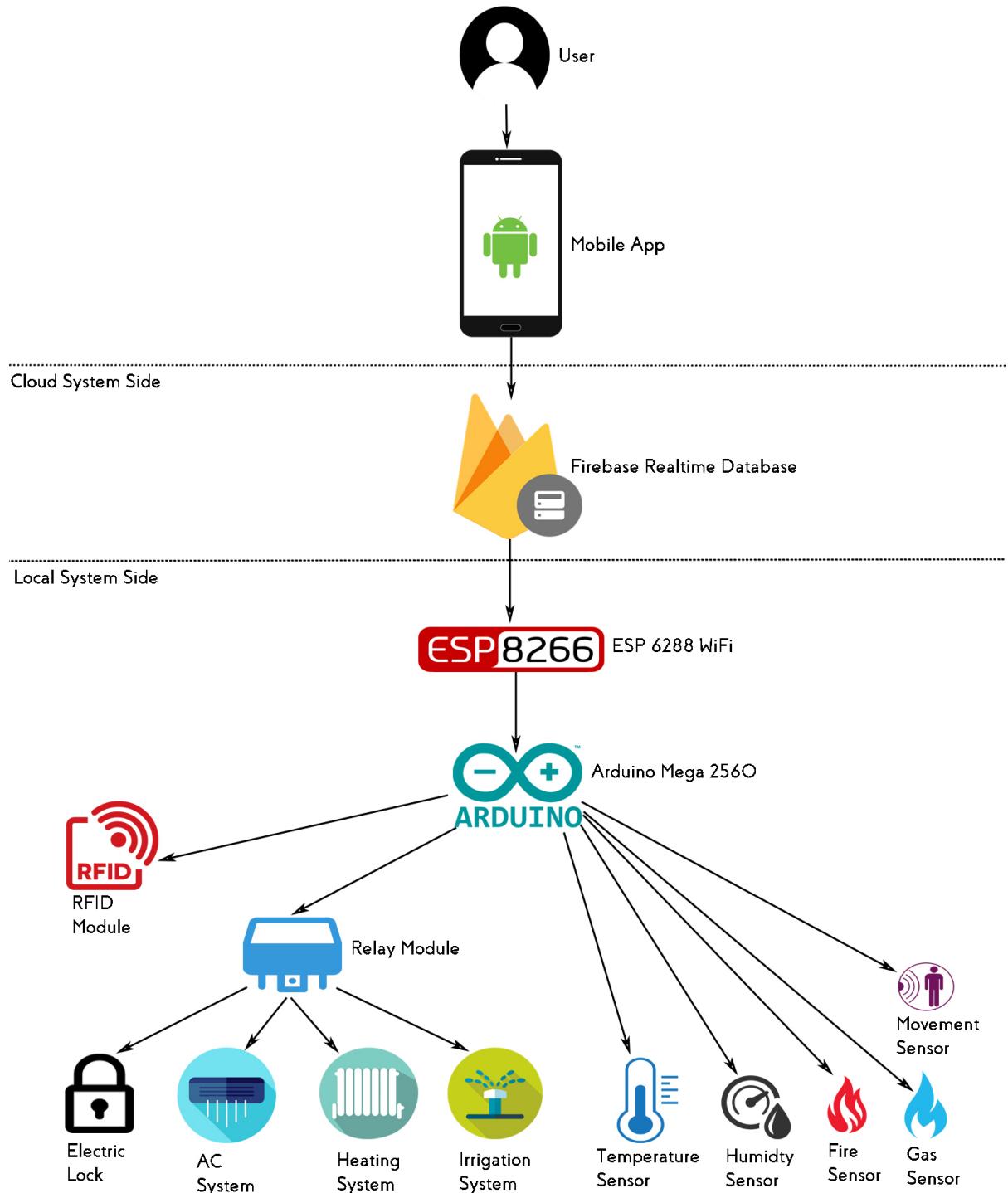


Figura 5.1: Arhitectura sistemului de monitorizare și automatizare al locuințelor

5.2 Descrierea vizuală a sistemului

În cadrul acestei secțiuni este prezentată vizual implementarea sistemului cu ajutorul diferitelor tipuri de diagrame. Astfel pentru reprezentarea tuturor funcționalităților disponibile este regăsită diagrama cazurilor de utilizare, iar pentru vizualizarea procesului de realizare al acestora sunt disponibile diagramele de secvență.

5.2.1 Diagrama cazurilor de utilizare

Diagrama cazurilor de utilizare prezentată în Figura 5.2 cuprinde toate funcționalitățile sistemului disponibile utilizatorului din cadrul aplicației mobile, incluzând acțiuni de monitorizare și automatizare a imobilului.

În cadrul diagramei se remarcă un număr de doi actori, și anume utilizatorul și sistemul local, aceștia aflându-se la capete opuse ale sistemului. Cel din urmă este responsabil cu procesarea locală a datelor, atât pentru primirea de comenzi cât și pentru trimiterea de date de monitorizare către utilizator prin intermediul serviciului de Cloud. De asemenea utilizatorul are la dispoziție două tipuri de cazuri de utilizare, și anume monitorizarea parametrilor din imobil și trimiterea de comenzi. Printre cazurile de monitorizare se remarcă vizualizarea temperaturii și detectia gazului metan, iar printre cazurile de trimisere a comenzi se remarcă controlul termostatului și controlul accesului în locuință.

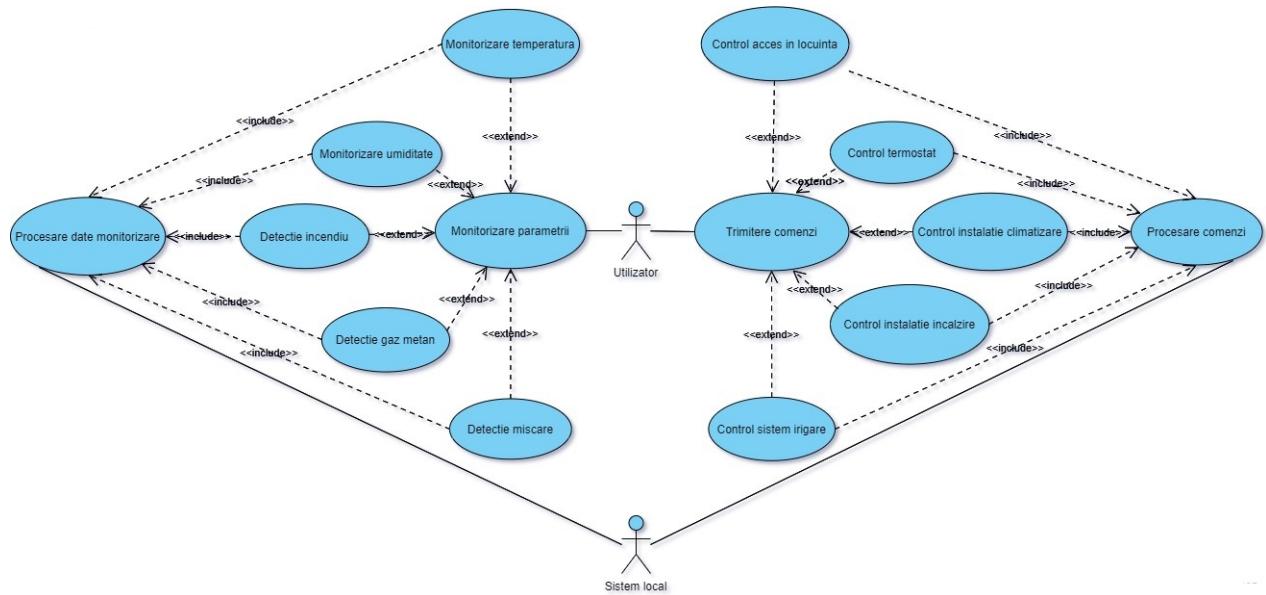


Figura 5.2: Diagrama cazurilor de utilizare a sistemului

5.2.2 Diagrame de secvență

Pentru ilustrarea procesului de monitorizare Figura 5.3 prezintă diagrama de secvență a vizualizării temperaturii din imobil în timp real. Logarea cu succes în cadrul aplicației este necesară, în caz contrar diagrama de secvență rămâneând la pasul ecranului de logare.

Procesul pentru monitorizarea celorlalți parametri are același mod de funcționare, diagrama având un caracter universal.

Diagrama de secvență a primirii alarmelor de urgență este similară cu cea din Figura 5.3, ca și diferență fiind atașat pe partea finală un senzor cu o întrerupere aferentă. Astfel actualizarea flag-ului de urgență se realizează în mod asincron, în cadrul unei întreruperi. Datele sunt actualizate și trimise în același mod către utilizator. Apare de asemenea un ecran nou de tipul unei notificări de urgență în momentul prelucrării alertei de către aplicația mobilă.

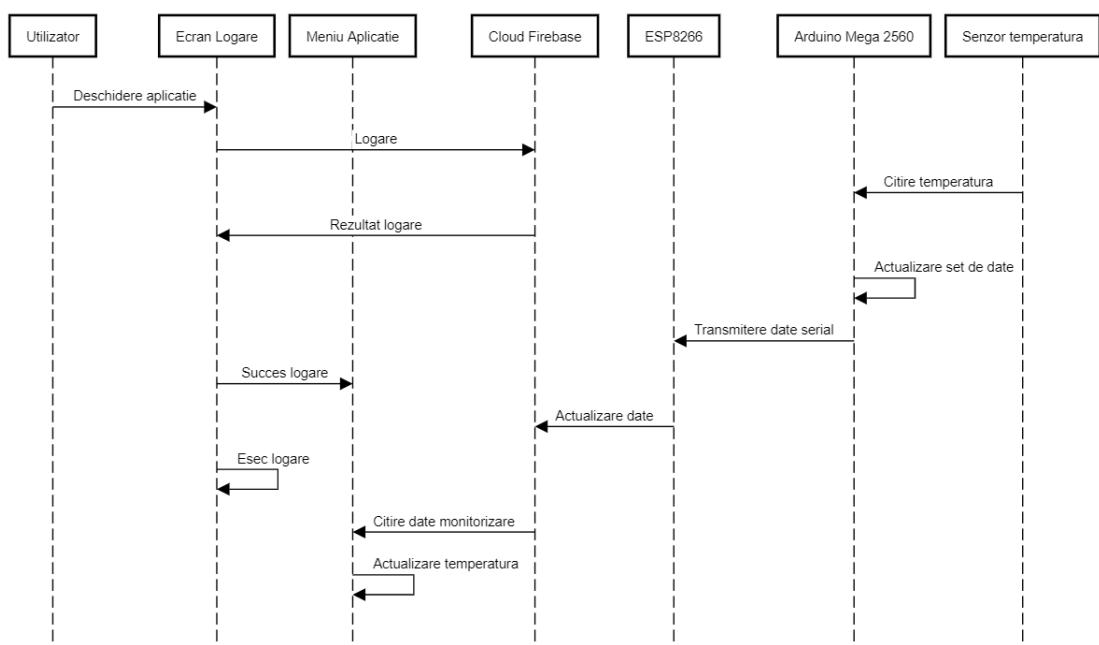


Figura 5.3: Diagrama de secvență a monitorizării temperaturii

Pentru ilustrarea procesului de realizare al unei comenzi Figura 5.4 prezintă diagrama de secvență a activării instalației de climatizare din imobil. Logarea cu succes în cadrul aplicației este necesară, în caz contrar diagrama de secvență rămâneând la pasul ecranului de logare.

Procesul pentru realizarea celoralte comenzi are un mod de funcționare asemănător, diferențele apărând la programarea termostatului/instalației de irigare unde este necesară trimiterea unei anumite valori. Pentru actualizarea în interfață grafică a aplicației mobile a statusului dispozitivelor comandate are loc procesul prezentat în Figura 5.3.

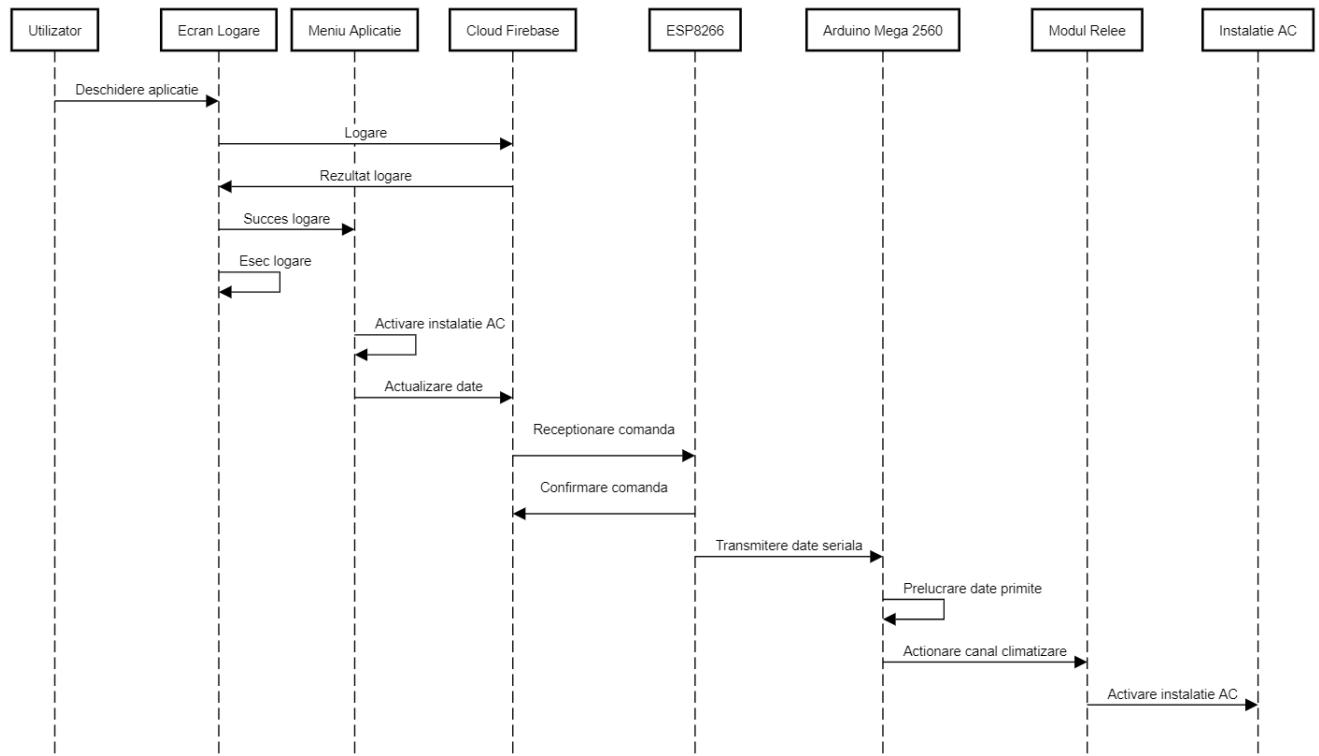


Figura 5.4: Diagrama de secvență a activării instalației de climatizare

5.3 Implementarea sistemului local

În cadrul acestei secțiuni este prezentată implementarea părții hardware a sistemului de monitorizare și automatizare a locuințelor, cuprinzând componentele utilizate, realizarea funcționalităților software și alte detalii de implementare.

5.3.1 Componentele necesare implementării

Sistemul hardware este alcătuit dintr-un număr de componente conectate la placa de dezvoltare Arduino Mega 2560, aceasta dispunând de un microcontroller integrat necesar pentru procesarea datelor pe partea locală. De asemenea modulul de wireless ESP 8266 are un microcontroller integrat, acesta comunicând în permanentă cu placa de dezvoltare menționată anterior cu ajutorul interfeței seriale. Pe lângă aceste două componente se regăsesc și diverse module, senzori și actuatori specifice. Acestea sunt prezentate în continuarea subsecțiunii.

Modulele regăsite în cadrul părții locale sunt următoarele:

- Tastatură matriceala 3x4 destinată interacțiunii utilizatorului cu sistemul local;

- Display LCD cu adaptor I2C destinat afișării meniului sistemului local;
- Buzzer Piezo destinat redării sunetelor și alarmelor locale;
- Modul de citire carduri RFID pentru scanarea cardului de acces în imobil;
- Modul de relee cu 4 canale pentru controlul actuatorilor/dispozitivelor din imobil;

Senzorii regăsiți în cadrul părții locale sunt următorii:

- Senzor de temperatură și umiditate DHT11 pentru monitorizarea în timp real;
- Senzor de detecție flăcări de tipul NIR cu întrerupere atașată pinului aferent;
- Senzor de detecție gaz metan MQ-4 cu întrerupere atașată pinului aferent;
- Senzor de detecție mișcare de tipul PIR cu întrerupere atașată pinului aferent;

Actuatorile/dispozitivele controlate din imobil compatibile cu sistemul Smart Home sunt următoarele:

- Ială electrică conectată la canalul 1 al modulului de relee;
- Pompă irigare conectată la canalul 2 al modulului de relee;
- Instalație climatizare conectată la canalul 3 al modulului de relee;
- Instalație încălzire conectată la canalul 4 al modulului de relee;

5.3.2 Schema electrică a părții locale

Schema electrică a părții hardware este prezentată în Figura 5.5, fiind indicat modul corect de realizare al conexiunilor locale. Placa de dezvoltare Arduino este alimentată de o sursă externă de 9V și 1A. Modulele externe sunt conectate în majoritate la o alimentare de 5V, în timp ce modulul de citire carduri RFID funcționează la o tensiune de 3.3V.

Modulul de wireless ESP 8266 este conectat la o sursă de curent de 5V individuală, având un consum de energie crescut și o sensibilitate ridicată la eventualele fluctuații ale curentului. În cazul fluctuațiilor modulul de wireless se resetează, astfel fiind pierdute date de transfer în urma întreruperii conexiunii.

Senzorul de detecție gaz metan MQ-4 are de asemenea un consum de energie crescut în etapă de pre-încălzire, această durând 5 minute din momentul pornirii sistemului. Cu toate acestea nu este necesară conectarea la o sursă separată, posibilele fluctuații ale curentului neavând un impact negativ asupra modulului.

Respectarea conexiunilor din cadrul schemei electrice este mandatorie, decuplarea anumitor module sau conectarea greșită a acestora ducând la funcționarea defectuoasă a funcțiilor aferente din sistem.

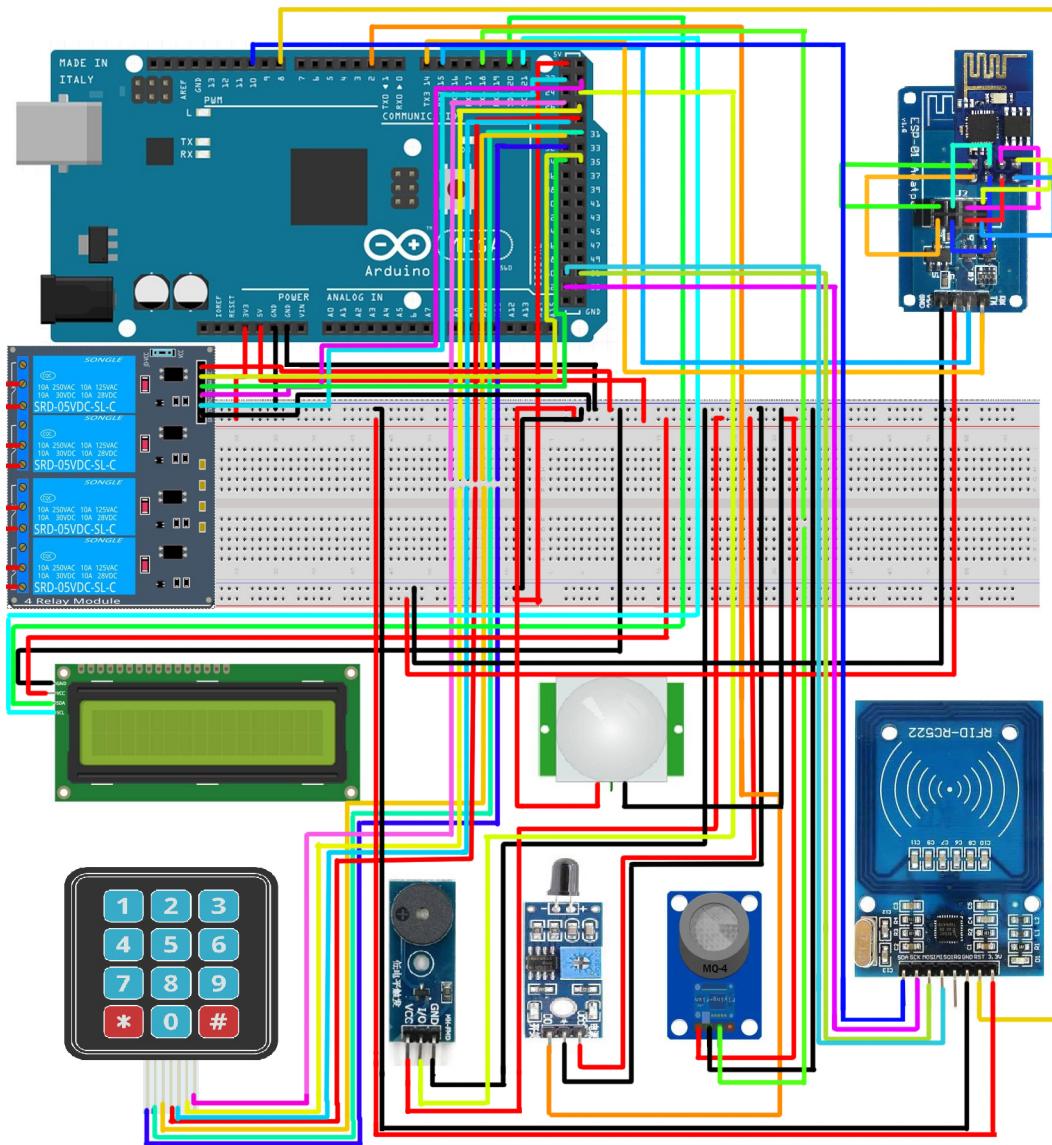


Figura 5.5: Schema electrică a sistemului local

5.3.3 Librării utilizate

Programarea plăcii de dezvoltare Arduino Mega 2560 se realizează separat de programarea modulului de wireless ESP 8266. Astfel pentru fiecare microcontroller se folosesc un set de biblioteci externe, librăriile utilizate pentru programarea plăcii de dezvoltare Arduino Mega fiind următoarele:

- SimpleDHT cu headerul SimpleDHT.h, utilizată pentru citirea temperaturii și a umidității de la senzorul DHT11;

- Wire cu headerul Wire.h, utilizată pentru comunicarea I2C cu LCD-ul integrat;
- LiquidCrystal I2C cu headerul LiquidCrystal_I2C.h, utilizată pentru controlul LCD-ului integrat cu ajutorul interfeței I2C;
- SPI cu headerul SPI.h, utilizată pentru comunicarea SPI cu cititorul de carduri RFID;
- MFRC522 cu headerul MFRC522.h, utilizată pentru procesarea cititorului de carduri RFID MFRC522;
- AVR Interrupts cu headerul avr/interrupt.h, utilizată pentru atașarea de întreruperi pinilor senzorilor responsabili cu alertele de urgență;
- Keypad cu headerul Keypad.h, utilizată pentru procesarea tastaturii 3x4 integrată în sistem.

Pentru programarea modului de wireless cu microcontroller integrat ESP 8266 sunt utilizate următoarele biblioteci:

- ESP8266 WiFi cu headerul ESP8266WiFi.h, utilizată pentru conectarea la o rețea wireless a modulului ESP 8266;
- Firebase Realtime Database ESP8266 cu headerul FirebaseESP8266.h, utilizată pentru conexiunea la o instanță a serviciului de Cloud Firebase Realtime Database;

5.3.4 Meniul sistemului

Sistemul local dispune de un meniu navigabil, cu șapte pagini prestabilite. Acesta este afișat în cadrul LCD-ului integrat, care comunică cu sistemul cu ajutorul interfeței I2C. Pentru navigare și interacționarea cu paginile meniului se utilizează tastatura matriceală 3x4, conectată la 7 pini ai plăcii de dezvoltare, 3 pentru coloane și 4 pentru rânduri.

Funcția responsabilă pentru afișajul meniului pe LCD este print_lcd(). Paginile sunt accesate în cadrul unui switch care depinde de parametrul lcd_page. Valorile acestuia cuprinse între 0 și 6 cuprind paginile navigabile de către utilizator, în timp ce valorile cuprinse între 7 și 9 cuprind paginile pentru alertele de urgență care nu îi sunt accesibile acestuia. Afișajul paginii actuale se realizează cu ajutorul funcțiilor clear(), setCursor() și print() din cadrul librăriei LiquidCrystal I2C. Funcția clear() curăță tot ce a fost afișat pe ecran de la pagina anterioară, în timp ce funcția setCursor() setează poziția de început pentru scrierea pe display, iar funcția print() afișează conținutul dorit la poziția selectată. Meniul este actualizat în timp real cu ajutorul variabilelor globale aferente diferenților parametri ai sistemului. Actualizarea se realizează la fiecare iterație a programului principal.

Funcția responsabilă de procesarea tastaturii matriceale și implicit interacțiunea directă cu meniul sistemului este keyboard_func(). Cu ajutorul funcției getKey() din cadrul

librăriei Keypad se obține tasta apăsată de utilizator. Se analizează valoarea acesteia și se actualizează parametrii necesari controlului meniului. Pentru navigarea paginilor se incrementează parametrul lcd_page la apăsarea tastei *, acesta fiind resetat la 0 în momentul în care ajunge la 6. În cadrul introducerii temperaturii dorite pentru funcția de termostat sistemul așteaptă acționarea unui număr mai mare de taste, acestea fiind procesate simultan în urma operației de confirmare. Operația de confirmare utilizată pentru termostat și pentru armarea sistemului de climatizare are loc în momentul apăsării tastei #.

5.3.5 Controlul accesului în locuință

Sistemul local dispune de o ială electrică care controlează accesul în cadrul imobilului. Aceasta este acționată de primul canal al modulului de relee integrat în sistem.

Funcția care se ocupă cu permiterea accesului în imobil pe partea locală este lock(), aceasta depinzând de parametrii right_card și lock_open pentru a se executa cu succes. Parametrul lock_open arată starea curentă a ialei, în timp ce variabila right_card devine validă în momentul în care cititorul de carduri RFID integrat în sistem scanează un card înregistrat cu succes.

Funcția care se ocupă de actualizarea parametrului right_card este rfid(). Aceasta detectează apropierea unui nou card scanabil de modulul MFRC522 cu ajutorul funcției PICC_IsNewCardPresent() din cadrul librăriei MFRC522. În etapa următoare programul citește codul unic al cardului cu ajutorul funcției PICC_ReadCardSerial() din cadrul librăriei menționate anterior, utilizând interfața serială SPI. Codul este citit în mod secvențial, rezultatul fiind stocat într-un string local cu ajutorul concatenării iterative. Dacă codul citit este același cu cel înregistrat în sistem variabila right_card este validată. Apare și parametrul wrong_card, acesta fiind validat în cazul în care codul citit în urma scanării unui card RFID nu este același cu codul salvat în sistem. Variabila wrong_card este folosită și în cadrul funcției buzzer-ului pentru producerea unui sunet de eroare în momentul scanării unui card gresit.

După ce parametrul right_card devine valid funcția lock() acționează iala electrică cu un temporizator de două secunde. Se utilizează funcția digitalWrite() pentru acționarea realelor specific și se initializează contorul de timp cu ajutorul funcției millis(). În urma scurgerii timpului alocat procesului reuel care controlează iala electrică devine inactiv, eliminându-se astfel posibilitatea că aceasta să rămână deschisă. La finalul procesului se actualizează toate flagurile implicate în execuție, și anume lock_open, right_card și wrong_card.

5.3.6 Funcția de termostat

Sistemul local dispune de un senzor DHT11 care permite monitorizarea în timp real a temperaturii și a umidității din imobil. Pentru obținerea celor doi parametri se folosește funcția read() din cadrul librăriei SimpleDHT, funcție care asignează valori temperaturii

și umidității simultan. Valorile sunt convertite în mod automat în grade Celsius pentru temperatură și în procentaje pentru umiditate.

Funcția de termostat este integrată în sistem și poate fi accesată pe partea locală cu ajutorul meniului navigabil. Aceasta poartă numele de temperature_func(), iar parametrul de care depinde execuția acesteia este selected_temp. Parametrul stocă temperatura optimă setată de utilizator pentru termostat în grade Celsius, acțiune realizată din cadrul aplicației mobile sau din cadrul meniului local. Temperatura actuală este comparată la fiecare iterare cu variabila selected_temp, modulul de relee acționând instalațiile de control al temperaturii din imobil corespunzătoare.

În cazul în care temperatura actuală este mai mică decât temperatura setată termostatul va porni instalația de încălzire prin acționarea releului aferent și se va asigura că instalația de climatizare este oprită prin dezactivarea releului corespunzător. Dacă temperatura actuală este egală cu temperatura dorită termostatul se asigură că ambele instalații de control al temperaturii sunt opriate, dezactivând canalele 3 și 4 ale modulului de relee. Pentru cazul în care temperatura din imobil este mai mare decât temperatura setată termostatul se va asigura că instalația de încălzire este oprită și va porni instalația de climatizare, dezactivând și activând cele două relee aferente. Pentru schimbarea stării releeelor de pe canalele 3 și 4 ale modulului integrat se folosește funcția digitalWrite(). Flagurile corespunzătoare funcționării sistemelor de control al temperaturii sunt actualizate la fiecare iterare pentru posibilitatea monitorizării, acestea fiind AC_on și heater_on.

5.3.7 Funcția de irigare

Partea locală dispune de un sistem de irigare programabil, o pompă de apă submersivă legată la al doilea canal al modulului de relee integrat. Acționarea acesteia se realizează doar din cadrul aplicației mobile, meniul local oferind doar informații despre starea irigării și timpul rămas în secunde până la terminarea procesului.

Funcția care se ocupă cu procesul de irigare este irrigation_func(), parametrul de care depinde execuția acesteia fiind irrigation_on. În cazul validării acestuia se stabilește inițial timpul necesar procesului prin convertirea variabilei irrigation_duration din minute în milisecunde și stabilirea limitei de execuție. Rezultatul stocat în noua variabilă irrigation_timer este comparat la fiecare iterare cu timpul curent de execuție, și anume valoarea returnată de funcția millis(). Releul de pe al doilea canal al modulului integrat este acționat până când se ajunge la finalul procesului de irigare, și anume când timpul scurs de execuție este superior limitei setate. În continuare se actualizează flagurile specifice și se asigură oprirea instalației, și anume se actualizează irrigation_on și irrigation_on_copy la valoarea 0, cel din urmă fiind valid inițial pentru setarea duresei de desfășurare.

5.3.8 Detecția factorilor de risc

În cadrul sistemului există un număr de trei senzori specializați în detecția factorilor de risc, și anume senzor de flăcări, senzori de detectie gaz metan MQ-4 și senzor de detectie

mișcare PIR. Pentru o detecție rapidă a posibilelor pericole fiecare senzor are o întrerupere atașată pinului digital aferent, cu funcțiile aferente:

- pentru dectia focului - attachInterrupt(0,fire_detection,FALLING);
- pentru dectia gazului metan - attachInterrupt(5,gas_detection,FALLING);
- pentru detectia mișcării - attachInterrupt(1,presence_detection,RISING);

Este folosită funcția attachInterrupt din cadrul librăriei AVR Interrupts pentru atașarea de întreruperi pinilor senzorilor aferenți, senzorul de detectie flăcări și senzorul de detectie gaz metan fiind activi pe 0 în timp ce senzorul de detectie mișcare este activ pe 1.

Funcția principală care se ocupă de monitorizarea imobilului este detection_manager(). Aceasta controlează validarea flagurilor aferente detectiei factorilor de risc din imobil, și anume fire_detected, gas_detected și presence_detected. În cazul în care unul dintre acești parametri devine valid funcția declanșează o alertă de urgență specifică.

Pentru declanșarea alertei programul validează variabila alarm_on, variabilă necesară emiterii de notificări pe partea aplicației mobile și interpretată în cadrul funcției buzz(), funcție care se ocupă de partea sonoră a acesteia pe partea locală. Este de asemenea actualizat parametrul new_page astfel încât să permită afișarea pe LCD-ul integrat a unui mesaj de urgență specific. Variabila lcd_page este setată la numărul paginii aferente detectiei actuale, număr cuprins între valorile 7 și 9.

Alarma de urgență activată continuă să fie activă pe partea locală până la dezarmarea acesteia de către utilizator. Pentru dezarmare în cadrul funcției detection_manager() se regăsește variabilă disarm_alarm, variabilă validată în cadrul funcției keyboard_func() prin apăsarea uneia dintre tastele * sau #. În urma opririi alarmei se revine la ecranul inițial prin actualizarea flagurilor folosite anterior. Altfel se atribuie valori noi variabilelor precum new_page, lcd_page, alarm_on, fire_detected, gas_detecte, presence_detected și disarm_alarm.

5.3.9 Comunicarea cu serviciul de Cloud

Pentru realizarea comunicării cu serviciul de Cloud a aplicației locale se utilizează modulul de wireless ESP 8266, acesta necesitând conexiunea permanentă cu placa de dezvoltare Arduino Mega 2560 prin intermediul interfeței seriale. Modul de trimitere și respectiv primire de date de la Cloud este prezentat în cadrul subsubsecțiunilor următoare.

Trimitere de date spre serviciul de Cloud

Pe partea plăcii de dezvoltare Arduino Mega datele sunt pregătite pentru a fi trimise spre modulul de wireless la fiecare două secunde, lucru realizat în cadrul funcției pushData(). Parametrii necesari monitorizării sunt convertiți la formatul String, urmând să

fie concatenați în mod secvențial la șirul de caracter destinat trimiterii. Acesta are ca și format delimitarea elementelor cu ajutorul separatorului , (virgulă urmată de un spațiu liber). După formarea șirului de date necesar trimiterii funcția initializează un temporizator, acesta fiind numit timer_send. Când cele două secunde s-au scurs cu succes, datele sunt trimise la modulul ESP 8266 prin intermediul interfeței seriale hardware cu numărul 3.

În cadrul modulului de wireless șirul cu datele monitorizare este citit în mod secvențial în cadrul buclei condiționată de funcția Serial.available(). Primirea completă a datelor este marcată de validarea parametrului data_received. Șirul cu informațiile de monitorizare este trimis la serviciul Firebase Realtime Database din Cloud cu ajutorul funcțiilor setString() și pushString() din cadrul librăriei Firebase Realtime Database ESP8266. Mai jos este să sunt prezentate ramurile pe care se inserează datele în cadrul bazei de date:

- Ramura origine - System Side/;
- Ramura date actuale - /Temperature,Humidity,FireDetected,GasDetected, PresenceDetected, Security, AC, Heating, Irrigation;
- Ramura pentru stocare date - /Stored Values;

Primirea de date de la serviciul de Cloud

Primirea de date de la serviciul Google Firebase se realizează în cadrul modulului de wireless ESP 8266. Aceasta verifică disponibilitatea de date noi pe canalul aferent prin utilizarea funcției getString din cadrul librăriei Firebase Realtime Database ESP8266. Dacă s-au găsit instanțe noi acestea sunt citite de modulul wireless sub formă unui string, urmând ca acesta să valideze primirea comenzi prin popularea canalului aferent cu valori de 2. Valoarea 2 a fost aleasă deoarece nu interacționează cu valorile de 0 și 1 ale flagurilor, fiind în același timp în afară câmpului de valori pentru parametrii cu informații. Mai jos este prezentată ramura utilizată pentru primirea de comenzi din cadrul bazei de date:

- Ramura origine - App Side/;
- Ramura date actuale - /Temperature,AC,Heating,Irrigation,Security,Lock;

Șirul de date citit este apoi trimis către placa de dezvoltare Arduino Mega prin intermediul interfeței seriale integrate. Datele sunt recepționate de microcontrollerul ATmega2560 cu ajutorul funcției receiveData(), în cadrul buclei condiționate de funcția Serial.available(). Datele sunt primite cu ajutorul interfeței seriale hardware cu numărul 3. Dacă sirul primit nu este null se folosesc funcțiile indexOf() și substring() pentru obținerea mai multor substringuri separate inițial de separatorul , (virgulă urmată de un spațiu liber). Acestea sunt convertite apoi în valori numerice aferente parametrilor din sistem responsabili cu execuția comenziilor de la utilizator. Noile valori obținute sunt interpretate în continuare pentru schimbarea stării actuatoarelor/dispozitivelor conectate la sistemul local. Modulul de relee se acționează cu ajutorul funcției digitalWrite().

5.4 Implementarea aplicației mobile

În cadrul acestei secțiuni este prezentată implementarea aplicației mobile destinată sistemului de monitorizare și automatizare al locuințelor, cuprinzând diagrama de clase a aplicației, analiza modulelor și alte detalii de implementare. Aplicația a fost dezvoltată cu ajutorul framework-ului Flutter, framework construit de cei de la Google pe baza limbajului Dart.

5.4.1 Diagrama de clase

Diagrama de clase a aplicației mobile este prezentată în Figura 5.6, fiind subliniate legăturile dintre clase și pachetele de care acestea aparțin. În cadrul diagramei se remarcă clasele Authentication, FirebaseAuth, Login, LoginHandler, MyApp, _MonitoringState, Monitoring, DatabaseReference, Timer și TempHumidTemplate.

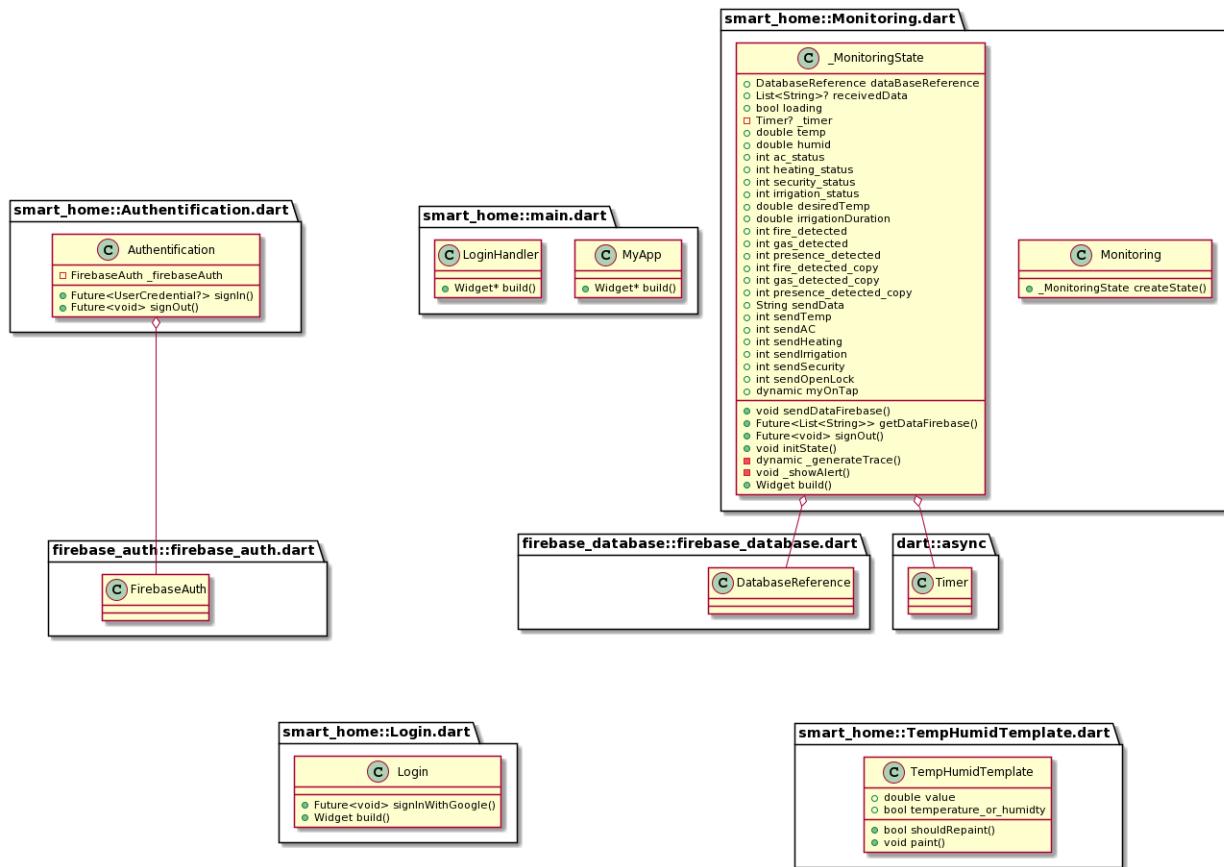


Figura 5.6: Diagrama de clase a aplicației mobile

Clasele Authentication, FirebaseAuth, Login și LoginHandler sunt folosite în cadrul procesului de logare în contul de Google aferent utilizatorului. Contul este legat la o instanță a bazei de date hostate cu ajutorul serviciului de Cloud Firebase.

MyApp este modulul destinat rulării aplicației, fiind clasa principală regăsită în fișierul main.dart. În cadrul ei se initializează procesul de autentificare, urmat de meniul aplicației în urma conectării cu succes.

Clasele _MonitoringState, Monitoring, DatabaseReference, Timer și TempHumidTemplate sunt responsabile cu funcționarea meniului aplicației. Acestea actualizează datele de monitorizare în timp real, oferind o conexiune permanentă la serviciul de Cloud Firebase cu scopul primirii și trimiterii de date noi.

5.4.2 Structura bazei de date

Pentru realizarea comunicării între partea locală a sistemului și aplicația mobilă este utilizată o instanță a serviciului Firebase Realtime Database, aceasta fiind prezentată în Figura 5.7. Se folosește astfel o bază de date NoSQL de tipul Real-time, parametrii necesari monitorizării fiind actualizați în timp real.

În cadrul serviciul de Cloud sunt prezente două canale principale de comunicare, și anume App Side pentru trimiterea de comenzi dinspre aplicația mobilă spre partea locală și System Side, canal utilizat pentru transmiterea parametrilor de monitorizare în timp real către aplicație.

În cadrul transmiterii de date de la partea locală spre aplicația mobilă se regăsesc două canale separate, și anume canalul /Stored Values pentru inserarea informațiilor primite și canalul datelor actualizate în timp real. Ambele canale se regăsesc sub ramură System Side/ și conțin String-uri obținute în urma concatenării parametrilor cu separatorul , (virgulă urmată de spațiu liber). Canalul destinat transmiterii de comenzi conține 6 câmpuri concatenate într-un singur String cu ajutorul separatorului , (virgulă urmată de spațiu liber) și este regăsit sub ramură App Side/

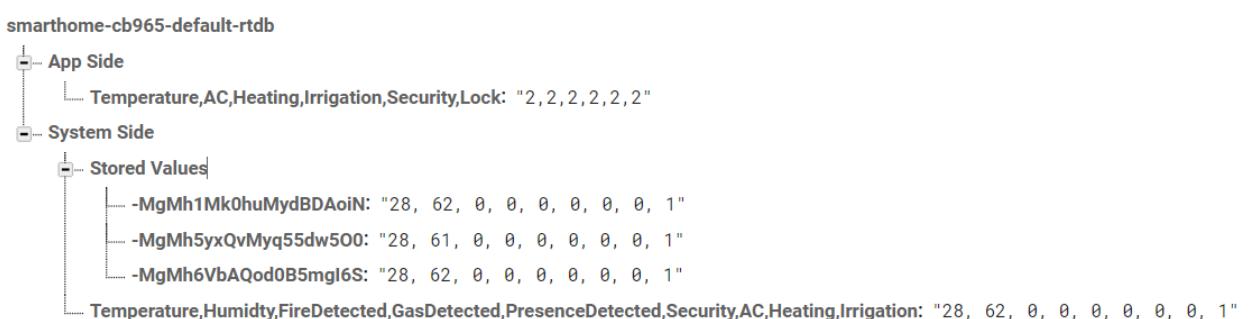


Figura 5.7: Structura bazei de date de tipul Real-time utilizată în proiect

Spațiul de stocare disponibil în cadrul unei instanțe a serviciului Firebase Realtime Database este de 1GB. Canalul destinat comunicării dintre partea locală și cea de Cloud conține ramura /Stored Values care stochează la fiecare secundă parametrii actuali primiți. Sireul aferent are o dimensiune de 27 octeți, oferind durată de funcționare continuă de maxim 428. Când limita de 1Gb a fost atinsă baza de date se va resetă, toate câmpurile fiind aduse la valoarea predefinită.

Fluxul de date din cadrul serviciul de Cloud este de maxim 40 octeți pe secundă, scenariu în care inserarea unui set nou de parametrii de monitorizare se suprapune cu trimiterea unui comenzi de către aplicația mobilă. Acest aspect este datorat dimensiunii câmpului cu parametrii de monitorizare actualizați de 27 bytes și a câmpului de comenzi cu dimensiune variabilă între 12 și 13 octeți.

În continuare sunt prezentate limitele de stocare în baza de date a seturilor de parametrii monitorizați pe parcursul anumitor perioade de timp, având în vedere funcționarea permanentă a sistemului:

- 86400 seturi de date de monitorizare pe parcursul unei zile, cu o dimensiune de 2.33 Mb alocată stocării;
- 604800 seturi de date de monitorizare pe parcursul unei săptămâni, cu o dimensiune de 16.32 Mb alocată stocării;
- 2629743 seturi de date de monitorizare pe parcursul unei luni, cu o dimensiune de 71 Mb alocată stocării;
- 31556926 seturi de date de monitorizare pe parcursul unui an, cu o dimensiune de 812.57 Mb alocată stocării;

Instanța bazei de date Firebase Realtime Database oferă de asemenea o limită de descărcare a datelor de 10 GB pe luna. Presupunând utilizarea constantă a aplicației mobile pentru monitorizare în timp real se utilizează 71 Mb din totalul de 10 GB destinați descărcării, restul de 9.92 Gb fiind destinați transmiterii de comenzi în timp real către partea locală. Având dimensiune maximă a setului de date cu toate flag-urile necesare procesării comenziilor de 13 octeți este posibilă transmiterea lunată a acestora în număr de 763076923.

5.4.3 Librării utilizate

În cadrul aplicației mobile sunt utilizate diverse librării pentru realizarea diferitelor funcționalități, acestea fiind regăsite în cadrul fișierului pubspec.yaml. Librăriile compatibile cu limbajul de dezvoltare Dart regăsite în proiect sunt următoarele:

- firebase_database cu versiunea 7.1.1, utilizată pentru realizarea conexiunii la baza de date de tipul Real-time hostată de serviciul de Cloud Firebase

- firebase_core cu versiunea 1.3.0, utilizată pentru realizarea conexiunii la serviciul de Cloud Firebase
- firebase_auth cu versiunea 2.0.0, utilizată pentru realizarea procesului de autentificare în cadrul instanței serviciul de Cloud Firebase
- vibration cu versiunea 1.4.0, utilizată pentru realizarea vibrațiilor fizice ale dispozitivului mobil în cadrul alertelor de urgență
- google_sign_in cu versiunea 5.0.5, utilizată pentru realizarea procesului de logare în contul de Google aferent sistemului
- provider cu versiunea 5.0.0, utilizată pentru confirmarea procesului de logare
- flutter_signin_button cu versiunea 2.0.0, utilizată pentru butoanele de logare și delogare din contul de utilizator, oferind o temă standardizată
- flutter_launcher_icons cu versiunea 0.9.0, utilizată pentru customizarea iconiței aplicației mobile

5.4.4 Logarea în contul de utilizator

Autentificarea în contul de utilizator aferent sistemului se realizează cu ajutorul serviciilor oferite de Google, și anume cu ajutorul clasei GoogleSignIn din cadrul librăriei google_sign_in și FirebaseAuth din cadrul librăriei firebase_auth.

Procedeul de logare este regăsit în clasa Authentication, în cadrul metodei signIn(), metodă asincronă cu argumentul returnat de tipul Future<UserCredential?>. Se asteaptă logarea utilizatorului cu ajutorul serviciilor menționate anterior și se generează un token unic. Mai jos este prezentat modul de instantiere al serviciilor de logare Google și obținerea unui utilizator valid:

```
final googleSignIn = GoogleSignIn();  
final googleUser = await googleSignIn.signIn();
```

Procedeul de părăsire a contului de utilizator este surprins în cadrul metodei asincrone signOut(), metodă unde se apelează deconectarea pe instanțele googleSignIn și _firebaseAuth create anterior.

Interfața grafică a paginii aferente procesului de logare este implementată în cadrul clasei Login, oferind metoda asincronă signInWithGoogle() pentru accesarea funcționalităților oferite de clasa prezentată anterior. Butonul care declanșează procesul de autentificare este plasat în interiorul unui ListView și este de tipul SignInButton, fiind importat din cadrul librăriei flutter_signin_button.

Clasa LoginHandler este instantiată în cadrul clasei MyApp din fișierul main.dart al proiectului, fiind încapsulată într-un obiect de tipul MultiProvider importat din librăria provider. Providerul verifică existența unei instanțe de logare în cont existente deja pe

dispozitiv, în caz contrar fiind afișat meniul de Login cu ajutorul clasei LoginHandler. Dacă utilizatorul este deja autentificat din cadrul unei sesiuni de utilizare trecute, clasa LoginHandler instantanează direct clasa Monitoring, aceasta fiind responsabilă de afișarea meniului principal.

Pentru posibilitatea realizării logarii de succes cu ajutorul serviciilor Google este nevoie de generarea certificatelor de amprentă SHA 1 și SHA-256. Acestea necesită înregistrarea în cadrul serviciului de Cloud Firebase și activarea opțiunii de logare pe baza contului Google.

5.4.5 Monitorizarea parametrilor

În urma autentificării cu succes aplicația mobilă oferă utilizatorului posibilitatea de monitorizare a diferenților parametrii din imobil în timp real. Interfața conține simboluri sugestive pentru fiecare parametru, acestea fiind înconjurate de arce de cerc cu culori sugestive a stării actuale.

Pentru primirea datelor de la serviciul de Cloud clasa _MonitoringState utilizează o referință la o instanță a clasei FirebaseDatabase din cadrul librăriei firebase_database. Utilizând metoda asincronă getDataFirebase() se obține o listă cu toți parametrii necesari monitorizării, aceasta fiind apelată la fiecare 1000 milisecunde cu ajutorul metodei _generateTrace și a timerului t. Datele sunt astfel stocate în variabile globale la fiecare secundă a execuției, fiind astfel accesibile tuturor elementelor din cadrul interfeței grafice. Mai jos este prezentat modul de citire al datelor de pe o ramură din cadrul bazei de date de tipul Real-time:

```
await DataBaseReference.child(..)
```

Aplicația are o interfață grafică derulabilă, cu toate elementele monitorizării incluse în cadrul aceleasi pagini, în mod succesiv. Se folosește astfel un ListView care cuprinde un număr semnificativ de widget-uri destinate reprezentării grafice a funcționalitatilor sistemului. Widget-urile utile monitorizării sistemului sunt prezentate în continuare.

Widget-urile de temperatură și umiditate sunt reprezentate cu ajutorul şablonului implementat în cadrul clasei TempHumidTemplate, cu metodă de afișare paint(). Aceasta primește ca și parametrii valoarea necesară afișării și flag-ul care indică dacă este necesară reprezentarea temperaturii sau a umidității. Valoarea maximă de reprezentare a temperaturii este 40, în timp ce pentru umiditate se remarcă valoarea 100. Cele două widget-uri se actualizează în mod automat la fiecare secundă datorită fluxului nou de date.

Widget-urile aferente sistemului de climatizare, sistemului de încălzire, sistemului de irigare și sistemului de securitate au același mod de comportare, fiind înconjurate de arce de cerc cu culori aferente stării lor actuale. Culorile utilizate sunt Colors.greenAccent pentru starea activă și Colors.red pentru starea inactivă. Culoarea arcului de cerc este condiționată de variabila globală aferentă parametrului monitorizat, actualizarea fiind astfel făcută la fiecare secundă.

5.4.6 Trimiterea de comenzi

Trimiterea de comenzi de pe partea aplicației mobile spre serviciul de Cloud se realizează cu ușurință din cadrul interfeței grafice, fiind disponibile butoane capacitive cu pictograme sugestive care arată starea actuală a dispozitivelor/sistemelor comandate.

În urma interacționării cu sistemele de comandă din cadrul aplicației mobile variabilele globale sunt actualizate și pregătite pentru trimiterea spre serviciul de Cloud. Funcția care realizează transmiterea datelor este `sendDataFirebase()`, concatenând parametrii cu ajutorul separatorului , (virgulă urmată de spațiu liber). După ce s-a obținut un String cu toate datele se inserează în cadrul bazei de date Firebase Realtime Database cu ajutorul referinței utilizate și pentru primirea de date. În continuare este prezentat modul de încărcare al datelor în cadrul unei ramuri existente în Cloud:

```
dataBaseReference.child(...).set(data);
```

Aplicația are o interfață grafică derulabilă, cu toate elementele de control incluse în cadrul aceleiași pagini, în mod succesiv. Se folosește astfel un `ListView` care cuprinde un număr semnificativ de widget-uri destinate interacțiunii utilizatorului cu sistemul. Widget-urile destinate trimiterii de comenzi sunt prezentate în continuare.

Widget-ul regăsit în partea superioară a interfeței este cel aferent controlului accesului în locuință. Acesta este reprezentat de un buton capacativ cu o pictogramă sugestivă, apăsarea acestuia declanșând modificarea parametrului `sendOpenLock` și apelarea funcției `sendDataFirebase()`.

Pentru funcția de termostat interfața grafică conține două widget-uri separate, și anume o bară cu valori pentru alegerea temperaturii și un buton pentru confirmarea temperaturii dorite. Temperatura selectată în cadrul bării este vizibilă și în cadrul textului butonului, fapt datorat utilizării variabilei globale `desiredTemp`. În momentul apăsării butonului temperatura aferentă termostatului este actualizată pentru trimitere și este apelată funcția `sendDataFirebase()`.

Modificarea stării de funcționare a instalațiilor de control a temperaturii se realizează cu ajutorul unui widget dedicat, un buton capacativ înconjurat de un arc de cerc cu culoare reprezentativă a stării de funcționare. Apăsarea acestuia rezultă în modificarea stării parametrilor globali aferenți climatizării sau încălzirii, fiind apelată apoi funcția `sendDataFirebase()`.

Pentru sistemul de irigare sunt disponibile trei widget-uri amplasate apropiat în interfață grafică. Acestea sunt butonul cu pictograma reprezentativă utilizat pentru comandarea sistemului de irigare directă, un buton cu textul aferent programării cu parametru a procesului și o bară cu valori pentru alegerea duratei de irigare dorite. Butonul principal este utilizat pentru controlarea fără durată a instalației, în timp ce butonul amplasat inferior apelează funcția `sendDataFirebase()` în urma setării duratei dorite. Procesul de setare al variabilelor globale este același ca și în cazul celorlalte widget-uri.

Ultimul widget destinat controlului de la distanță regăsit în cadrul aplicației mobile este cel aferent sistemului de securitate. Prin apăsarea acestuia se vaarma/dezarma sis-

temul de securitate, și anume se va actualiza valoarea variabilei globale aferente în cadrul funcției onPressed() și se va apela funcția sendDataFirebase().

5.4.7 Alertă de urgență

Aplicația mobilă dispune de trei tipuri de alertă, două de urgență și una de securitate. Acestea sunt detectia de flăcări în interiorul imobilului, detectia scurgerilor de gaz metan și detectia mișcării în urma armării sistemului de securitate.

Funcția care se ocupă de declanșarea alertelor de urgență pe partea aplicației mobile este _showAlert(), oferind că și feedback o vibrație inițială de o secundă. Aceasta este implementată cu ajutorul funcției vibrate() din cadrul librăriei vibration. Pentru fereastra specifică alertelor se utilizează un obiect de tipul AlertDialog, fiind ales un fundal roșu cu text în majuscule. Titlul alertei și textul acesteia sunt transmiși ca și parametrii, fiind stabiliți în urma primirii și analizei setului de date de la Cloud.

Pentru confirmarea primirii alertei este necesară atingerea ecranului dispozitivului mobil în orice loc, acest aspect însă nefiind suficient pentru stoparea alarmei pe partea locală. Aceasta necesită dezactivarea manuală cu ajutorul tastaturii integrate local.

Capitolul 6

Testare și Validare

Pentru procesul de testare și validare al sistemului Smart Home au fost analizate atât partea locală cât și aplicația mobilă în relație cu cea anterioară. S-au folosit metode de testare precum testarea manuală și cea automată, aspect ales în mod avantajos în funcție de partea sistemului discutată. Rezultatele obținute au fost satisfăcătoare, oferind un nivel de acuratețe ridicat al monitorizării.

6.1 Testarea sistemului local

Sistemul local a fost testat în mod manual, fiind utilizat monitorul serial integrat în cadrul mediului de dezvoltare Arduino IDE. Configurațiile alese pentru testare au fost următoarele:

- Pentru placa de dezvoltare Arduino Mega 2560 a fost ales un baud rate de 9600 biți pe secundă;
- Pentru modului de wireless ESP-8266 a fost ales un baud rate de 115200 biți pe secundă.

Testarea plăcii de dezvoltare Arduino Mega 2560 alături de componentele aferente a fost una locală, folosindu-se un string compus din toți parametrii necesari monitorizării, string folosit în cadrul implementării pentru trimiterea de informații în timp real către serviciul de Cloud. Formatul acestuia din urmă este “Temperature, Humidity, FireDetected, GasDetected, PresenceDetected, Security, AC, Heating, Irrigation”.

Valorile afișate în timp real pe lcd-ul integrat au fost comparate cu stringul printat în cadrul monitorului serial . De asemenea a fost urmarită evoluția parametrilor de monitorizare pe parcursul cazurilor reale de utilizare, obținându-se un delay nesemnificativ de actualizare a datelor. Activarea dispozitivelor locale/actuatorilor a fost de asemenea analizată cu ajutorul modulului de relee integrat. Toți parametrii afișați în terminal exceptând temperatura și umiditatea reprezintă valori booleene, 1 reprezentând statusul activ

al elementului respectiv. În Figura 6.1 este prezentat modul de monitorizare al datelor din cadrul monitorului serial pentru realizarea procesului de testare.

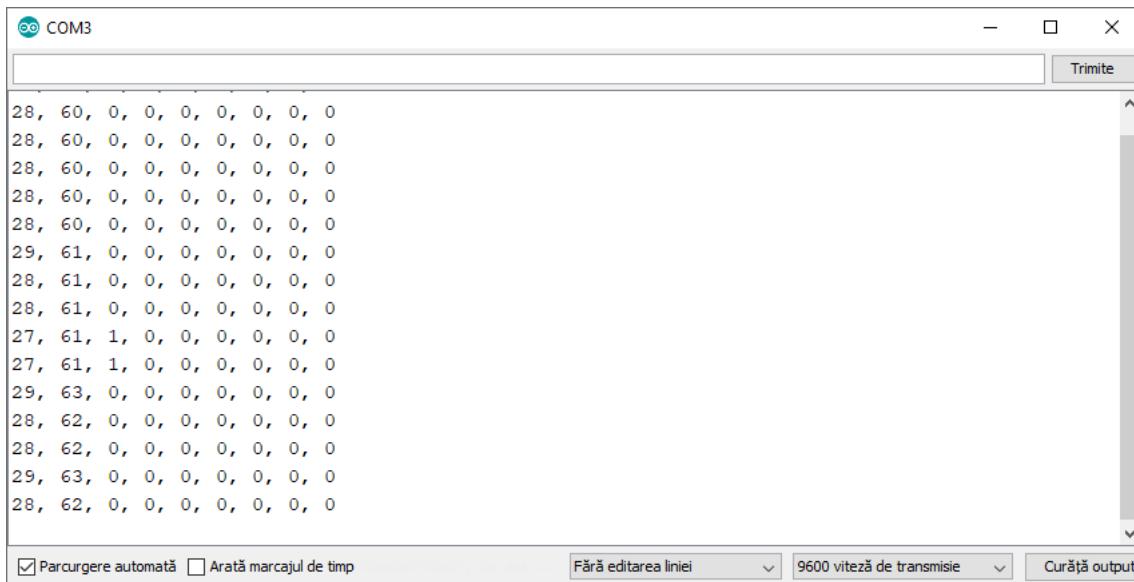


Figura 6.1: Utilizarea monitorului serial pentru testarea sistemului local

În cadrul testării sistemului local se realizează și testarea individuală a modulului de wireless ESP-8266 cu ajutorul monitorului serial integrat în IDE. Prin folosirea firm-ware-ului AT se poate testa funcționarea corectă a modulului și conectarea acestuia cu succes la o rețea locală de internet. În urma conexiunii de succes se testează funcționarea corectă a serviciilor oferite de instanța de Firebase aferentă. Se urmărește ca inserarea în baza de date Real-time să nu difere de rezultatele transmise de pe partea locală și secțiunea de comenzi primită de la aplicația mobilă să fie recepționată și resetată cu succes de către modulul ESP.

6.2 Testarea aplicației mobile

Testarea aleasă pentru partea aplicației mobile este una automată, folosindu-se teste care analizează rezultatul primit în urma efectuării unei comenzi. Astfel se folosește ultimul set de date inserat în baza de date de tipul Real-time de către partea locală, se trimit o comandă de pe partea aplicației mobile și se așteaptă un timp predefinit pentru rezultatul dorit. Dacă în intervalul de timp predefinit în baza de date se inserează un set de date satisfăcător, testul returnează succes. Canalele bazei de date utilizate pentru procesul de testare sunt următoarele:

- “System Side/Stored Values” pentru primirea datelor inițiale și finale;

- “App Side/Temperature,AC,Heating,Irrigation,Security,Lock” pentru trimiterea comenzii.

Pentru procesul de testare al aplicației mobile s-au efectuat un numar de 100 teste automate realizate cu ajutorul limbajului Python, fiind obținută o rată de succes de peste 90%. Cazurile nefavorabile sunt datorate în mare parte vitezei reduse a conexiunii la internet a aplicației mobile/sistemului local, fapt care cauzează o latență crescută și implicit încheierea testului curent prin depășirea timpului de execuție. În Figura 6.2 sunt prezentate seturile de date din cadrul bazei de date de tipul Real-time înaintea și respectiv după trimiterea unei comenzi de pe partea aplicației mobile.



Figura 6.2: Testarea seturilor de date primite în urma trimiterii unei comenzi

Testarea manuală poate fi utilizată în secțiunea aplicației mobile pentru verificarea funcționării elementelor din interfața grafică, verificând modificările apărute în cadrul seturilor de date stocate cu ajutorul serviciului Firebase în urma interacționării cu aplicația. Se testează astfel afișarea în timp real a datelor, trimiterea de comenzi și primirea de alerte în aplicație. Acet mod de testare este asemănător cu cel utilizat pentru evaluarea partii locale, fiind folosite direct datele din baza de date de tipul Real-time pentru validare, fără a fi nevoie de fișiere de testare.

Capitolul 7

Manual de Instalare și Utilizare

7.1 Ghid de instalare

Partea locală a sistemului de Smart Home este bazată pe placa de dezvoltare Arduino Mega 2560 Rev3, bazată pe microcontrollerul ATmega2560 oferit de cei de la compania Atmel. Toate celelalte componente interconectate sunt compatibile cu microcontrollerul menționat anterior.

Pentru programarea și depanarea părții locale a sistemului este nevoie de instalarea mediului de dezvoltare Arduino IDE pe un computer care rulează ca sistem de operare Windows sau Mac OS X. Pentru programarea plăcii de dezvoltare Arduino Mega 2560 se folosește mediul de dezvoltare menționat anterior, fiind necesară instalarea următoarelor librării adiționale din secțiunea Instrumente> Gestionaare bibliotecii:

- Keypad
- SimpleDHT
- MFRC522
- LiquidCrystal I2C

Pe parcursul procesului de dezvoltare se selectează din secțiunea Instrumente> Manager Plăci opțiunea Arduino Mega sau Mega 2560 pentru programarea microcontroller-ului principal. Modulul de wireless utilizat de sistem este programat separat, fiind nevoie de pași adiționali precum:

- Instalarea din cadrul secțiunii Instrumente> Manager de Plăci a modulului ESP8266;
- Instalarea bibliotecii adiționale Firebase ESP8266 Client;
- Selectarea plăcii Generic ESP8266 Module din secțiunea Instrumente> Manager de Plăci.

Pentru încărcarea codului pe placa de dezvoltare Arduino Mega 2560 se folosește interfața serială USB, selectându-se un port disponibil. În cazul modulului de wireless ESP-8266 încarcarea programului se realizează tot cu ajutorul unei plăci de dezvoltare Arduino. Se realizează conectarea pinilor specifici și se încarcă programul în urma selectării modulului de placă specific. Mai jos sunt menționate conexiunile necesare pentru scrierea programului în memoria modulului de wireless:

- GND,GPIO-0 > GND
- GPIO-2,RST > neconectat
- RXD > RX
- TXD > TX
- CHPD,VCC > 3.3V

Pentru partea de Cloud a sistemului este necesară crearea unei instanțe de Firebase, serviciu oferit de cei de la Google. Contul asociat cu noua instanță va fi folosit și pentru logarea în cadrul aplicației mobile aferente sistemului. Pentru comunicarea din Cloud se va folosi baza de date de tipul Real-time oferită de cei de la Firebase. Cele două canale de comunicare se generează automat în momentul conectării celorlalte două parti ale sistemului la internet, nefind nevoie să fie inițializate de dezvoltator sau utilizator.

Aplicația mobilă destinată utilizatorului necesită mediul de dezvoltare Android Studio IDE pentru rulare și depanare, cu versiunea 2020.3.1 sau una mai nouă disponibilă. Mediul de dezvoltare menționat anterior este destinat instalării pe sistemele care rulează unul dintre următoarele OS-uri: Windows, MAC OS X, Ubuntu sau Chrome OS. Aplicația este dezvoltată cu ajutorul framework-ului Flutter, lansat de cei de la compania Google, framework bazat pe limbajul de programare Dart.

În urma instalării mediului de dezvoltare Android Studio cu o versiune compatibilă este necesară instalarea framework-ului Flutter. Se descarcă arhiva de pe site-ul oficial al celor de la Flutter, se extrage și se copiază calea către folderul bin pentru a putea fi adăugată ca și variabilă de mediu a sistemului de operare. Pentru verificarea instalării corecte a framework-ului menționat anterior se utilizează comanda următoare într-un terminal nou deschis:

```
flutter doctor -v
```

Înaintea rulării aplicației mobile din cadrul mediului de dezvoltare Android Studio este necesară accesarea fișierului inclus în proiect cu numele pubspec.yaml și executarea comenții pub get, disponibilă în partea superioară a paginii. Astfel vor fi verificate și actualizate toate pachetele necesare rulării aplicației. În cazul în care codul returnat este 0 se poate rula cu succes aplicația. Este posibilă și folosirea comenții următoare în terminalul integrat în IDE:

```
dart pub get
```

Pentru a rula aplicatia este nevoie de un dispozitiv mobil cu o versiune de Android 10 sau mai nouă, conectat prin cadrul magistralei usb la statia de dezvoltare. O alta varianta este utilizarea unui emulator integrat în cadrul IDE-ului Android Studio, acesta putand fi configurat după preferințe, dar consumând mai multe resurse locale precum memorie RAM și spațiu de stocare. În urma compilării de succes o copie a aplicației este instalată pe dispozitivul mobil selectat, funcționalitățile putând fi testate în întregime.

7.2 Manual de utilizare

7.2.1 Utilizarea sistemului local

Pentru utilizarea partii locale a sistemului de monitorizare și automatizare a locuinței este nevoie de alimentarea acestuia cu alimentatorul de 9V și 1A inclus. În urma alimentării cu succes sistemul căută rețea de internet configurată în cadrul modulului de wireless și încearcă conectarea. Fără realizarea conexiunii dorite sistemul continuă să funcționeze, dar fără a putea trimite seturi de date spre Cloud sau primi comenzi de pe partea aplicației mobile. Dacă conexiunea s-a realizat cu succes atunci sistemul este complet funcțional și gata de folosit de la distanță.

Pasul următor este conectarea corectă a actuatorilor/dispozitivelor din interiorul imobilului la modulul dedicat de relee cu 4 canale. Dispozitivele compatibile satisfac limitele de 230V/10A pentru AC și 30V/10A pentru DC. În Figura 7.1 este prezentată conectarea corectă a dispozitivelor la releele aferente.

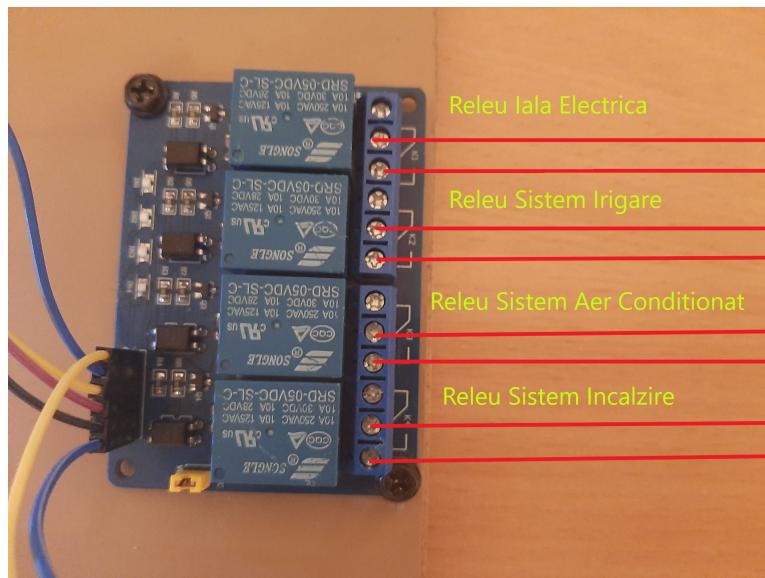


Figura 7.1: Mod de conectare actuatoare/dispozitive la sistem

Sistemul local oferă un număr restrâns de funcționalități afișate pe lcd-ul integrat, opțiuni care pot fi navigate și executate cu ajutorul tastaturii matriceale incorporate. Tastele cuprind toate cifrele și 2 simboluri pentru o navigare ușoară. Modul de utilizare al acestora este:

- Simbolul * este destinat navigării succesive a meniurilor;
- Simbolul # este destinat acțiunii de selectare/confirmare;
- Cifrele 0-9 sunt destinate introducerii de date.

Sistemul vine însotit de 7 pagini de meniu navigabile în mod succesiv, cu diverse informații. După cum a fost menționat și anterior, trecerea de la un meniu la altul se realizează cu ajutorul tastei *. În Figura 7.2 sunt surprinse toate paginile meniului disponibile utilizatorului, în ordinea lor de navigare.



Figura 7.2: Paginile de meniu disponibile

- Prima pagină a meniului afișează în timp real temperatura în grade celsius și umiditatea sub formă de procentaj.

- Cea de a doua pagina conține setarea temperaturii dorite pentru funcția de termostat. Prin introducerea unei temperaturi între 10 și 29 de grade celsius urmată de apăsarea tastei # se setează temperatura dorită în imobil, operația putând fi oricând abandonată prin apăsarea tastei *.
- Cea de a treia pagină conține informații în timp real despre securitatea imobilului, cum ar fi starea ialei electrice și starea sistemului de alarmă.
- A patra pagină conține opțiunea de aarma sau adezarma securitatea locuinței prin apăsarea tastei #.
- A cincea pagină prezinta în timp real starea sistemelor de aer condiționat și incălzire din locuință.
- Cea de a șasea pagină arată în timp real starea sistemului de irigare aferent imobilului, cu timpul rămas în secunde până la terminare.
- Ultima pagină conține statusul conexiunii la internet a sistemului local, alături de timpul de funcționare exprimat în secunde.

Sistemul local vine cu un cititor de carduri RFID integrat pentru accesul în locuință. Prin scanarea cardului corect de către utilizator iala electrică principală se deschide cu un anumit temporizator, închizându-se automat peste două secunde pentru securitatea imobilului. În cazul scanării unui card RFID neînregistrat sistemul emite un sunet de alertă. În Figura 7.3 este prezentat cazul de utilizare al cititorului de carduri RFID integrat în sistem.

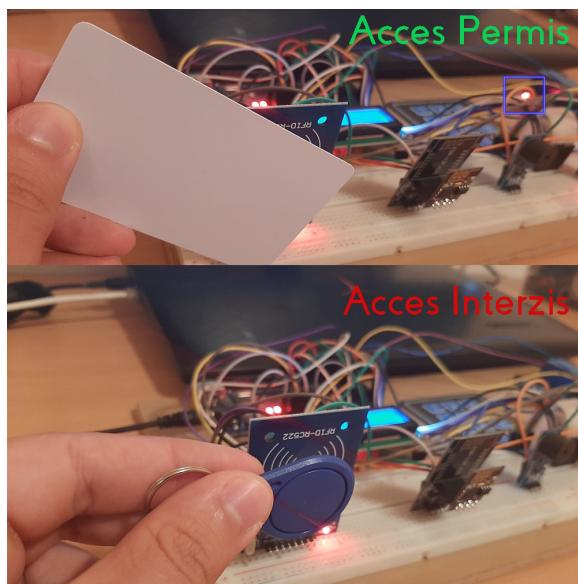


Figura 7.3: Cititorul de carduri RFID integrat

Există 3 tipuri de alerte, 2 alerte de urgență și o alertă de securitate. Acestea sunt alerta de incendiu, alerta de surgeri gaz metan și alerta de detecție mișcare în imobil. Alerta de mișcare se poate declanșa doar dacă utilizatorul a armat securitatea sistemului și senzorul PIR detectează un intrus în interiorul imobilului.

Alertele declanșate cauzează o alarmă de urgență pe partea locală, alarme care vin însoțite de 3 pagini specifice suplimentare meniului navigabil afișate pe ecranul lcd. Alaramele pot fi dezactivate doar de pe partea locală a sistemului de către utilizator prin apăsarea tastelor * sau #.

7.2.2 Utilizarea aplicației mobile

Pentru utilizarea aplicației mobile este necesară folosirea unui dispozitiv mobil cu o versiune de Android 10 sau mai nouă, cu conexiune mandatorie la internet în momentul utilizării. Aplicația poate fi instalată cu ajutorul unui fișier cu extensia .apk, fiind compatibilă cu orice dispozitiv care îndeplinește condițiile menționate anterior.

Prima pagină a aplicației mobile este pagina de Login. Accesul în cont se face strict pe baza contului Google aferent instanței de Firebase folosite pentru sistemul de Smart Home. În urmăre autentificării cu un cont valid utilizatorul are acces la toate funcționalitățile aplicației de monitorizare. Procesul de autentificare este prezentat în Figura 7.4.

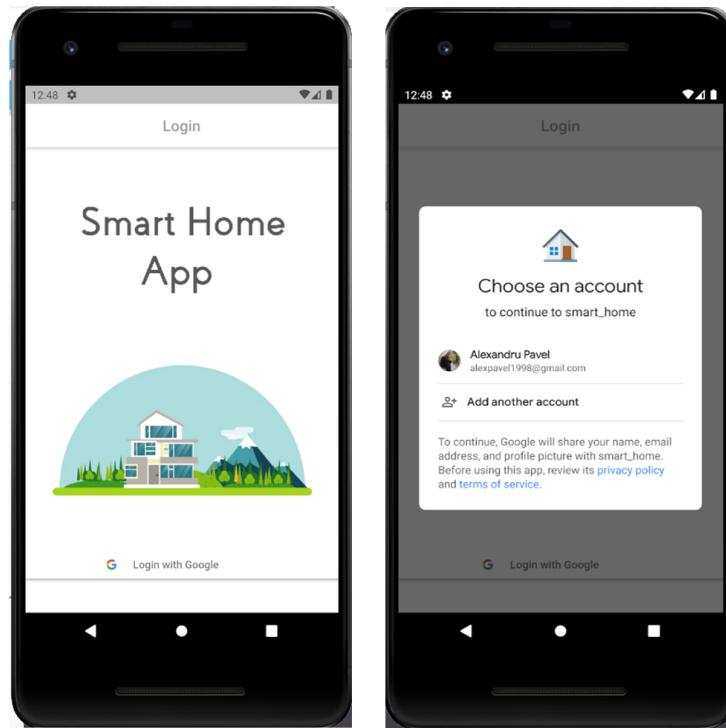


Figura 7.4: Autentificare în aplicație cu contul Google aferent

Dacă autentificarea în cont a fost cu succes, utilizatorul are acces la tot panoul de monitorizare al sistemului de Smart Home. Toate funcțiile sunt incluse într-o singură pagină derulabilă, fapt care oferă fluiditate și viteză de utilizare. Prima funcționalitate prezentă în pagină este deblocarea ialor rezidenței, fiind cea mai accesibilă și des folosită de utilizator. Prin apăsarea butonului de deblocare iala se deschide cu un temporizator, închizându-se peste aproximativ două secunde din motive de securitate.

Următoarea secțiune a aplicației mobile este monitorizarea în timp real a temperaturii și a umidității, pentru cea dintâi existând funcția de termostat. Astfel utilizatorul poate selecta temperatură dorită cu ajutorul unei bări ajustabile, cu temperaturi cuprinse între 10 și 29 grade celsius, marcată cu câte un punct la fiecare prag pentru o utilizare ușoară. După selectarea temperaturii dorite, utilizatorul apasă un simplu buton și comanda este trimisă spre partea locală a sistemului. Funcționarea termostatului se poate observa în continuarea aplicației, la alternarea funcționării instalației de aer condiționat cu cea de încălzire. Setarea temperaturii dorite la 17 grade celsius poate fi observată în Figura 7.5, alături de o temperatură actuală de 27 grade și o umiditate de 54%.

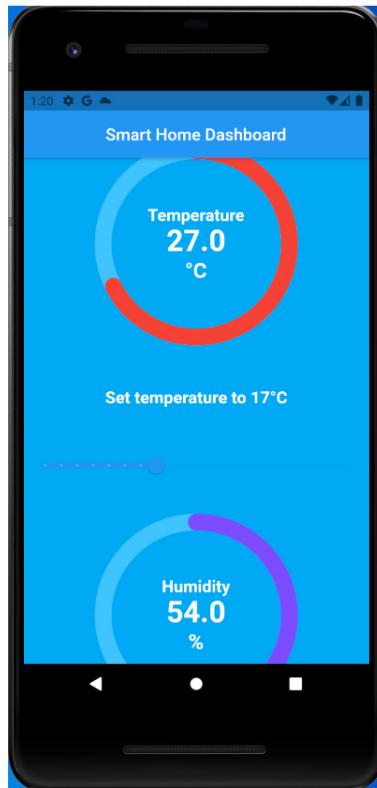


Figura 7.5: Temperatura, umiditatea și funcția de termostat a aplicației

Următoarea secțiune a aplicației reprezintă starea diverselor sisteme și actuatorilor din cadrul imobilului, cum ar fi instalația de aer condiționat și instalația de încălzire a

rezidenței. Pentru a schimba starea acestora este nevoie de o singură apăsare. Culoarea roșie în jurul butonului indică lipsa de activitate a componentei, în timp ce culoarea verde reprezintă că aceasta este activă. Dacă se dorește schimbarea stării instalației de climatizare sau încălzire în timp ce termostatul este activat, activitatea acestuia este suspendată iar noua comandă se execută. Cele două instalații de control al temperaturii nu pot lucra simultan, fapt care una cere anularea celeilalte. În Figura 7.6 este prezentat exemplul în care încalzirea este activă iar utilizatorul apasă butonul de climatizare.

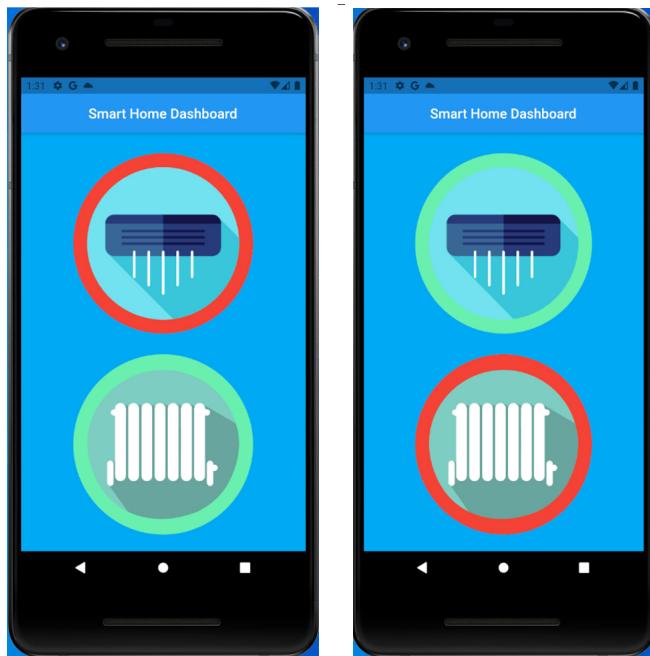


Figura 7.6: Acționarea dispozitivelor de control al temperaturii

Secțiunea care urmează în interfața aplicației este destinată sistemului de irigare, acesta putând fi pornit sau oprit oricând din aplicație. De asemenea se poate seta durata de irigare la orice moment printr-un sistem asemănător cu cel al termostatului, cu durată între 5 și 30 de minute. După selectarea duratei cu ajutorul bării ajustabile prin apăsarea butonului aferent irigarea va funcționa până la scurgerea timpului sau până la apăsarea din nou a butonului. Valoarea prestabilită a irigării este de 10 minute.

Ultima secțiune a aplicației este butonul pentru armarea sau dezarmării securității. Culoarea roșie indică faptul că securitatea imobilului nu este activă, astfel se vor primi doar alertele de pericol. În momentul în care culoarea butonului este verde securitatea a fost armată, aplicația mobilă primind ambele tipuri de alerte. În caz de alertă aplicația mobilă blochează interfața generală a utilizatorului, acesta fiind avertizat în funcție de tipul de alertă cu un ecran roșu urmat de vibrația telefonului. Notificarea de alertă persistă până în momentul în care utilizatorul atinge ecranul, alarma locală continuând însă până este dezarmată din interiorul imobilului. În figura 7.7 avem un exemplu de alertă de securitate

cauzată în urma activării sistemului de monitorizare de către utilizator.

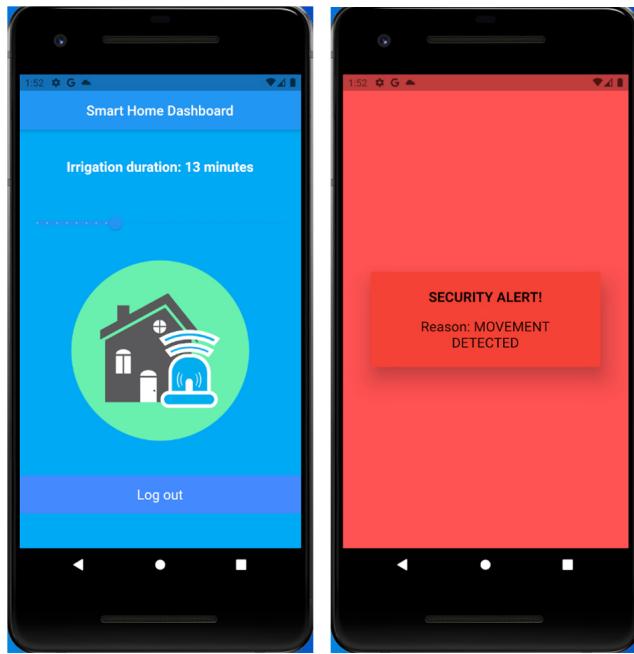


Figura 7.7: Alertă de securitate cauzată de detecția mișcării

Părăsirea contului actual din aplicație se face cu ajutorul unui buton regăsit în partea de jos a interfeței aplicației. În momentul apăsării acestuia utilizatorul este redirectat la prima pagină, anume cea de logare. Închiderea aplicației va duce la delogarea automată din cont. Aplicația mobilă suportă mai mulți utilizatori logați simultan în cadrul unui cont, funcție care este utilă pentru membrii familiei/colocatarii unui imobil.

Capitolul 8

Concluzii

8.1 Realizări și analiza rezultatelor obținute

Proiectul reușește să implementeze un ecosistem de automatizare și monitorizare al imobilelor, implementând majoritatea funcționalităților necesare regăsite pe piața sistemelor de Smart Home cu succes.

Partea locală a sistemului este pregătită pentru utilizare, toate componentele fiind deja integrate, regăsindu-se aspectul de Plug and Play. Funcționalitatile implementate au o acuratețe crescută în urma testărilor repetate, oferind posibilitatea de monitorizare și trimitere de comenzi în timp real. Deconectarea sistemului local de la rețeaua de internet nu împiedică monitorizarea și executarea anumitor comenzi din interiorul imobilului, pentru acestea existând posibilitatea de navigare cu ajutorul lcd-ului și a tastaturii integrate. Conectarea dispozitivelor/actuatorilor din imobil este facilitată cu ajutorul unui modul de relee, acestea putând fi adăugate sau înălțurate în orice moment de către utilizator. Comenzile primite de pe partea aplicației mobile sunt prelucrate cu succes de sistemul local, acuratețea fiind una ridicată în urma testelor efectuate. Sistemul local nu necesită mentenanță hardware sau software, ecosistemul functionând cu succes atât timp cat configurarea inițială rulează.

Partea de Cloud a sistemului este reprezentată de o bază de date de tipul Real-time hostată cu ajutorul serviciului Firebase oferit de cei de la Google. Acest aspect îi conferă fiabilitate și nivel de securitate ridicate, mentenanța nefiind necesară atât timp cat nu se interacționează direct cu datele stocate. Datele sunt trimise sub formă de String-uri, fapt care conferă o latență cât mai scazută la transmitere. De asemenea datele de monitorizare sunt stocate în cadrul bazei de date pentru posibilitatea de analiză ulterioară. Partea locală și aplicația mobilă au sisteme de verificare a formatului datelor primite de la Cloud, eliminându-se astfel posibile hazarduri exterioare.

+Aplicația mobilă este intuitivă și ușor de utilizat, logarea în contul aferent sistemului realizându-se cu ajutorul contului Google. Interfața cuprinde toate funcționalitățile într-o singură pagină derulabilă, oferind utilizatorului posibilitatea de a trimite comenzi multiple simultan. De asemenea informațiile utile monitorizării sunt actualizate în timp

real, deconectarea aplicației de la internet salvând în interfață ultimele date preluate. În urma procesului de testare s-a obținut o acuratețe ridicată a trimiterii cu succes de comenzi dinspre partea aplicației mobile către partea locală. În comparație cu partea locală, aplicația oferă toate funcționalitățile ecosistemului Smart Home, acestea fiind:

- Monitorizare parametri din imobil (temperatură/umiditate)
- Termostat (10-29 grade Celsius)
- Control dispozitive termice (instalație aer condiționat/incălzire)
- Control dispozitive din afara imobilului (instalație irigare)
- Armare/Dezarmare sistem securitate
- Recepționare alerte de urgență/securitate (incendiu, scurgeri gaz metan, intrus)

În urma unificării celor trei nivele și a funcționării simultane a acestora rezultă un sistem complet de automatizare și monitorizare al imobilelor, acesta fiind pregătit pentru utilizare. Rezultatele obținute în urma testării întregului sistem au fost satisfăcătoare, fiind posibilă adăugarea de funcționalități suplimentare pentru o experiență de utilizare mai complexă.

8.2 Dezvoltări și îmbunătățiri ulterioare

Piața sistemelor Smart Home este în continuă creștere, utilizatorii dorind periodic funcționalități noi pentru creșterea nivelului de trai. Având în vedere acest aspect, pentru o complexitate mai ridicată a sistemului se pot realiza următoarele dezvoltări și îmbunătățiri ulterioare:

- Adăugarea modulelor de relee suplimentare pentru posibilitatea controlării unui număr mai mare de actuatori/dispozitive din imobil;
- Extinderea meniului navigabil pe partea locală cu toate funcțiile disponibile și adăugarea unui display cu o complexitate mai ridicată;
- Folosirea unui modul NodeMCU pe partea locală pentru reducerea delay-ului de comunicație cu serviciul de Cloud și transmiterea datelor pe canale multiple;
- Adăugarea de secțiuni individuale/ramuri pentru fiecare parametru monitorizat în cadrul serviciului Firebase pentru analize ulterioare.
- Adăugarea de modalități suplimentare de logare în contul aferent aplicației mobile, cum ar fi logarea prin email și parolă, logarea prin Facebook, etc.;

- Sistem de repartiție pe camere a dispozitivelor în cadrul aplicației, fiecare cameră putând fi monitorizată individual în timp real;
- Secțiune cu date statistice integrată în aplicație, utilă pentru o monitorizare de ansamblu a rezidenței;
- Implementarea unui sistem de tipul Drag And Drop în cadrul aplicației mobile pentru mutarea elementelor în interfață și adăugarea de noi actuatori/dispozitive predefinite;
- Integrarea funcției de calendar și programarea diferitor acțiuni în cadrul anumitor intervale orare/zile din an;
- Adăugarea unui sistem de securitate GSM independent de aplicație pentru alertele de urgență, util în cazul lipsei de conexiune la internet.

Implementând funcționalitați din cadrul celor menționate mai sus complexitatea ecosistemului Smart Home va crește, oferind un grad de practicabilitate mai ridicat utilizatorului. Apariția unor tehnologii noi implică imbunătățirea funcționalitatilor deja implementate, acestea putând crește viteza de utilizare și acuratețea monitorizării.

Bibliografie

- [1] S. Kumar, P. Tiwari, and M. Zymbler, ‘Internet of things is a revolutionary approach for future technology enhancement: a review,’ *Journal of Big Data*, vol. 6, no. 111, 2019.
- [2] M. Lia, W. Gub, W. Chenc, Y. Hed, Y. Wud, and Y. Zhange, ‘Smart home: Architecture, technologies and systems,’ *Procedia Computer Science*, vol. 131, pp. 393–400, 2018.
- [3] K. Gatsis and G. J. Pappas, ‘Wireless control for the iot: power spectrum and security challenges,’ *IEEE/ACM second international conference on internet-of-things design and implementation (IoTDI)*, 2017.
- [4] J. Bartje, ‘The top 10 iot application areas – based on real iot projects,’ *IoT Analytics*, 2016. [Online]. Available: <https://iot-analytics.com/top-10-iot-project-application-areas-q3-2016/>
- [5] E. Seo, S. Baeb, H. Choib, and D. Choib, ‘Preference and usability of smart-home services and items - a focus on the smart-home living-lab,’ *Journal of Asian Architecture and Building Engineering*, 2020. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/13467581.2020.1812397>
- [6] N. David, A. Chima, A. Ugochukwu, and E. Obinna, ‘Design of a home automation system using arduino,’ *International Journal of Scientific Engineering Research*, vol. 6, no. 6, 2015.
- [7] M. Malathi, M. Dhanushyaa, A. Gowsalya, and A. Janani, ‘Home automation on esp8266,’ *International Journal of Computer Science and Engineering*, 2017. [Online]. Available: https://www.researchgate.net/publication/316926871_HOME_AUTOMATION_ON_ESP8266