

**Universitatea Tehnică a Moldovei**

**Sistem portabil pentru monitorizarea factorilor care influențează sănătatea**

**Portable system for monitoring factors influencing health.**

|  |  |
| --- | --- |
| **Student:** | **gr. TI-206,**  **Pleșu Cătălin** |
| **Coordonator:** | **Secrieru Adrian**  **asistent universitar** |

**Chişinău, 2024**

**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Admis la susţinere**

**Şef departament:**

**FIODOROV Ion dr., conf.univ.**

**--------------------------------**

**„\_\_\_”**  **2024**

**Sistem portabil pentru monitorizarea factorilor care influențează sănătatea**

**Proiect de licenţă**

|  |  |  |
| --- | --- | --- |
| **Student:** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Pleșu Cătălin, TI-206** |
| **Coordonator:** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Secrieru Adrian, asist. univ.** |
| **Consultant:** | **\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Cojocaru Svetlana, asist.univ.** |

**Chişinău, 2024**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii Tehnologia informaţiei**

**Aprob**

**Șef departament:**

**Fiodorov Ion, dr., conf.univ.**

**„ *06” octombrie* *2023***

**CAIET DE SARCINI**

**pentru proiectul de licenţă al studentului**

*Pleșu Cătălin*

*(numele şi prenumele studentului)*

**1. Tema proiectului de licenţă** *Sistem portabil pentru monitorizarea factorilor care influențează sănătatea*

**confirmată prin hotărârea Consiliului facultăţii nr.** *2* **din**  „ *06*” *octombrie*  *2022*

**2. Termenul limită de prezentare a proiectului** **de licență** *20.05.2023*

**3. Date iniţiale pentru elaborarea proiectului de licență** *Sarcina pentru elaborarea proiectului de diplomă.*

**4. Conţinutul memoriului explicativ**

*Introducere*

*1 Analiza domeniului de studiu*

*2 Scopul, obiectivele și cerințele sistemului*

*3 Modelarea și proiectarea sistemul informatic*

*4 Realizarea sistemului*

*5 Documentarea produsului realizat*

*6 Estimarea costurilor proiectului*

*Concluzii*

5. Conţinutul părţii grafice a proiectului de licență

[Figura 1.1 - Atmotube PRO 17](#Figura!0|sequence)

[Figura 1.2 - Garmin Tempe 17](#Figura!1|sequence)

[Figura 1.3 - DT-8820 18](#Figura!2|sequence)

[Figura 2.1 - modelul paginii de pornire, principale și setări 24](#Figura!3|sequence)

[Figura 2.2 - modelul paginii pentru alarme, configurări și detalii 25](#Figura!4|sequence)

[Figura 3.1 - Utilizarea sistemului 30](#Figura!5|sequence)

[Figura 3.2 - Vizualizarea datelor 30](#Figura!6|sequence)

[Figura 3.3 - Gestionarea sistemului 31](#Figura!7|sequence)

[Figura 3.4 - Conectarea la o platformă externă 32](#Figura!8|sequence)

[Figura 3.5 - Actualizarea versiunii software 33](#Figura!9|sequence)

[Figura 3.6 - Citirea parametrului 34](#Figura!10|sequence)

[Figura 3.7 - Evaluare alarme 35](#Figura!11|sequence)

[Figura 3.8 - Exportarea datelor 36](#Figura!12|sequence)

[Figura 3.9 - Stările dispozitivului 37](#Figura!13|sequence)

[Figura 3.10 - Sub-stările etapei de operațiune 37](#Figura!14|sequence)

[Figura 3.11 - Colectarea datelor 38](#Figura!15|sequence)

[Figura 3.12 - Conectarea unui nou dispozitiv 39](#Figura!16|sequence)

[Figura 3.13 - Colectarea datelor 40](#Figura!17|sequence)

[Figura 3.14 - Conectarea unui nou dispozitiv 40](#Figura!18|sequence)

[Figura 3.15 - Implementarea senzorilor 42](#Figura!19|sequence)

[Figura 3.16 - Gestionarea stărilor dispozitivului 43](#Figura!20|sequence)

[Figura 3.17 - Componentele aplicației mobile 44](#Figura!21|sequence)

[Figura 3.18 - Componentele software-ului dispozitivului 44](#Figura!22|sequence)

[Figura 3.19 - Implementarea sistemului 46](#Figura!23|sequence)

[Figura 4.1 - Pagina de pornire 48](#Figura!24|sequence)

[Figura 4.2 - Lista de dispozitive 49](#Figura!25|sequence)

[Figura 4.3 - Configurarea unei alarme 49](#Figura!26|sequence)

[Figura 4.4 - Setările aplicației 50](#Figura!27|sequence)

[Figura 4.5 - Pagina de detalii 50](#Figura!28|sequence)

[Figura 4.6 - Localizarea 51](#Figura!29|sequence)

[Figura 4.7 - Istoricul de mesaje 51](#Figura!30|sequence)

[Figura 6.1 - 53](#Figura!31|sequence)

**6. Lista consultanţilor:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Consultant** | **Capitol** | **Confirmarea realizării activităţii** | |
| **Semnătura consultantului (data)** | **Semnătura studentului (data)** |
| *Cojocaru Svetlana* | *Standarde tehnologice, Controlul calităţii, Estimarea costului proiectului* |  |  |

**7. Data înmânării caietului de sarcini** *01.09.2023*

**Coordonator** *Secrieru Adrian* **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*semnătura*

**Sarcina a fost luată pentru a fi executată de către studentul** *Pleșu Cătălin*

*01.09.2023*

*semnătura, data*

**PLAN CALENDARISTIC**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr. crt.** | **Denumirea etapelor de proiectare** | **Termenul de realizare a etapelor** | **Nota** |
| *1* | *Elaborarea sarcinii, primirea datelor pentru sarcină* | *02.09.22– 30.09.22* | *10%* |
| *2* | *Analiza domeniului de studiu* | *06.10.21– 30.11.21* | *15%* |
| *3* | *Proiectarea sistemului* | *01.12.21 – 25.12.21* | *15%* |
| *4* | *Realizarea sistemului* | *10.01.22 – 05.03.22* | *35%* |
| *5* | *Descrierea sistemului* | *06.03.22– 01.04.22* | *10%* |
| *6* | *Estimarea costurilor sistemului* | *02.04.22– 15.04.22* | *15%* |
| *7* | *Finisarea proiectului* | *16.04.22– 14.05.22* | *5%* |

**Student**  *Pleșu Cătălin ( )*

**Coordonator de proiect de licență** *Secrieru Adrian ( )*

**DECLARAŢIA STUDENTULUI**

Subsemnatul Pleșu Cătălin, declar pe proprie răspundere că lucrarea de faţă este rezultatul muncii mele, pe baza propriilor cercetări şi pe baza informaţiilor obţinute din surse care au fost citate şi indicate, conform normelor etice, în note şi în bibliografie. Declar că lucrarea nu a mai fost prezentată sub această formă la nici o instituţie de învăţământ superior în vederea obţinerii unui grad sau titlu ştiinţific ori didactic.

Semnătura autorului

**UNIVERSITATEA TEHNICĂ A MOLDOVEI**

**FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ**

**DEPARTAMENTUL INGINERIA SOFTWARE ȘI AUTOMATICĂ**

**PROGRAMUL DE STUDII TEHNOLOGIA INFORMAȚIEI**

**AVIZ**

la proiectul de licență

**Titlul:** Sarcina a fost luată pentru a fi executată de către studentul.

Studentul Pleșu Cătălin gr. TI-206

**1.** **Actualitatea temei:** Factorii mediului influențează direct sănătatea umană, mulți dintre acești factori nici măcar nu pot fi identificați cu ajutorul simțurilor umane. Tehnologiile IoT moderne permit crearea de dispozitive care să monitorizeze mediul și să alerteze utilizatorul în cazul când valoarea parametrilor iese din limitele dorite, ceea ce ar putea aduce efecte negative asupra sănătății.

**2.** **Caracteristica proiectului de licență**: Sistemul a fost creat pentru a facilita analiza parametrilor mediului din proximitatea utilizatorului.

**3.** **Analiza prototipului:** Aplicația mobilă este utilizată pentru a afișa datele colectate și pentru a interacționa cu dispozitivul.

**4. Estimarea rezultatelor obținute:** Acest sistem este capabil de a monitoriza parametrii mediului. Utilizarea aplicației pare a fi intuitivă însă asta nu este 100%.

**5. Corectitudinea materialului expus:**

**6. Calitatea materialului grafic:**

**7. Valoarea practică a proiectului:**

**8. Observaţii şi recomandări:**

**9. Caracteristica studentului şi titlul conferit:** \_\_\_\_\_\_\_\_\_

***Lucrarea în forma electronică corespunde originalului prezentat către susținere publică.*Coordonatorul proiectului de licenţă** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Secrieru Adrian, asist.univ**

*semnătura, data*

REZUMAT

Se scrie pe scurt din ce este compusă lucrarea cu specificarea și descrierea succintă a fiecărui capitol

ABSTRACT

Se scrie rezumatul în limba engleză.

CUPRINS

[ABREVIERI 10](#__RefHeading___Toc7954_2078331531)

[INTRODUCERE 13](#__RefHeading___Toc7956_2078331531)

[1 ANALIZA DOMENIULUI DE STUDIU 14](#__RefHeading___Toc7958_2078331531)

[1.1 Importanța temei 15](#__RefHeading___Toc7960_2078331531)

[1.2 Sisteme similare cu proiectul realizat 15](#__RefHeading___Toc7962_2078331531)

[1.3 Analiza tehnologiilor 18](#__RefHeading___Toc7964_2078331531)

[1.4 Parametrii mediului 20](#__RefHeading___Toc7966_2078331531)

[2 SCOPUL, OBIECTIVELE ȘI CERINȚELE SISTEMULUI 22](#__RefHeading___Toc7968_2078331531)

[2.1 Cerințe funcționale 23](#__RefHeading___Toc4861_3003112200)

[2.2 Cerințe non-funcționale 25](#__RefHeading___Toc4863_3003112200)

[3 MODELAREA ȘI PROIECTAREA SISTEMUL INFORMATIC 27](#__RefHeading___Toc7970_2078331531)

[3.1 Descrierea comportamentală a sistemului 28](#__RefHeading___Toc7972_2078331531)

[3.1.1 Imaginea generală asupra sistemului 29](#__RefHeading___Toc7974_2078331531)

[3.1.2 Modelarea vizuală a fluxurilor 33](#__RefHeading___Toc7976_2078331531)

[3.1.3 Stările de tranzacție a sistemului 36](#__RefHeading___Toc7978_2078331531)

[3.1.4 Descrierea scenariilor de utilizare a aplicaţiei 38](#__RefHeading___Toc7980_2078331531)

[3.1.5 Fluxurile de mesaje şi legăturile dintre componentele sistemului 39](#__RefHeading___Toc7982_2078331531)

[3.2 Descrierea structurală a sistemului 40](#__RefHeading___Toc7984_2078331531)

[3.2.1 Descrierea structurii statice a sistemului 42](#__RefHeading___Toc7986_2078331531)

[3.2.2 Relatiile de dependență între componentele sistemului 43](#__RefHeading___Toc7988_2078331531)

[3.2.3 Modelarea echipamentelor mediului de implementare 45](#__RefHeading___Toc7990_2078331531)

[4 REALIZAREA SISTEMULUI 47](#__RefHeading___Toc7992_2078331531)

[4.1 Descrierea la nivel de cod pe module 48](#__RefHeading___Toc7994_2078331531)

[4.1.1 Aplicația mobilă 48](#__RefHeading___Toc4865_3003112200)

[4.2 Testarea sistemului 51](#__RefHeading___Toc7996_2078331531)

[5 DOCUMENTAREA PRODUSULUI REALIZAT 52](#__RefHeading___Toc7998_2078331531)

[6 ESTIMAREA COSTURILOR PROIECTULUI 53](#__RefHeading___Toc8000_2078331531)

[CONCLUZII 54](#__RefHeading___Toc8002_2078331531)

[BIBLIOGRAFIE 55](#__RefHeading___Toc8004_2078331531)

[ANEXA A 57](#__RefHeading___Toc8006_2078331531)

ABREVIERI

* IoT (eng. Internet of Things) - internetul lucrurilor, rețeaua de dispozitive fizice interconectate care comunică și cooperează;
* IT (eng. Information Technology) - tehnologia informației, utilizarea calculatoarelor și a tehnologiei pentru stocarea, procesarea și transmiterea informațiilor;
* PM1 (eng. Particulate matter 1) - particule în suspensie cu diametrul de 1 micrometru sau mai mic, care pot afecta sănătatea umană,
* PM2.5 (eng. Particulate Matter 2.5) - particule în suspensie cu diametrul de 2.5 micrometri sau mai mic, care pot avea efecte nocive asupra sănătății umane;
* PM10 (eng. Particulate Matter 10) - particule în suspensie cu diametrul de 10 micrometri sau mai mic, care pot afecta calitatea aerului și sănătatea umană;
* VOCs (eng. Volatile Organic Compounds) - compuși organici volatili, substanțe chimice care pot emana vapori și care pot avea efecte nocive asupra sănătății umane;
* ANT+ (eng. Advanced and Adaptive Network Technology) - tehnologie de comunicație wireless utilizată în special în dispozitivele de fitness și sănătate pentru a transmite date către alte dispozitive compatibile;
* Wi-Fi (eng. Wireless Fidelity) - tehnologie de rețea fără fir care permite dispozitivelor să se conecteze la internet și să comunice între ele într-o rețea locală;
* FPGA (eng. Field-Programmable Gate Array) - matrice de porți programabilă, dispozitive hardware care pot fi reconfigurate după producție pentru a îndeplini diverse funcții;
* ASIC (eng. Application-Specific Integrated Circuit) - circuit integrat specific aplicației, o formă de cipuri electronice pecializate proiectate pentru o anumită aplicație;
* VHDL (eng. VHSIC (eng. Very High-Speed Integrated Circuit Hardware Description Language) Hardware Description Language) - limbaj utilizat pentru proiectarea circuitelor digitale;
* C - limbaj de programare de nivel înalt utilizat în dezvoltarea software-ului de sistem și de aplicații;
* C++ - limbaj de programare de nivel înalt și orientat pe obiecte, derivat din limbajul C, utilizat în special pentru dezvoltarea de software complex;
* OS (eng. Operating System) - sistem de operare, software-ul de bază care controlează hardware-ul unui computer și permite utilizatorilor să ruleze aplicații și să interacționeze cu sistemul;
* RTOS (eng. Real-Time Operating System) - sistem de operare în timp real, un tip de sistem de operare care oferă garanții de timp în executarea sarcinilor;
* OTA (eng. Over-the-Air) - actualizare prin aer, procesul de actualizare a software-ului dispozitivelor sau a echipamentelor fără a fi necesară o conexiune fizică;
* ZigBee - standard de comunicare fără fir utilizat în rețelele locale, în special în aplicații de automatizare a clădirilor și a casei inteligente;
* LoRaWAN (eng. Long Range Wide Area Network) - rețea de arie largă cu distanțe mari, tehnologie de comunicație wireless de lungă distanță și de consum redus de energie, folosită pentru conectarea dispozitivelor IoT la internet;
* AMQP (eng. Advanced Message Queuing Protocol) - protocol avansat de mesagerie de tip mesaj-coloană, pentru transmiterea mesajelor între aplicații;
* MQTT (eng. Message Queuing Telemetry Transport) - protocolul de transport al mesajelor mașină la mașină, utilizat în special pentru dispozitive IoT pentru a trimite și a primi date într-o manieră eficientă;
* HTTP (eng. Hypertext Transfer Protocol) - protocolul de transfer hipertext, utilizat pentru transmiterea informațiilor pe internet;
* CoAP (eng. Constrained Application Protocol) - protocolul constrâns de aplicație, pentru aplicațiile web de tip restful pe dispozitivele iot cu resurse limitate;
* DDS (eng. Data Distribution Service) - serviciul de distribuție a datelor, o tehnologie de middleware utilizată pentru transmiterea și schimbul de date între dispozitivele conectate la rețea;
* LwM2M (eng. Lightweight M2M) - protocol și servicii de administrare a dispozitivelor IoT utilizate pentru gestionarea și monitorizarea dispozitivelor IoT în rețelele LPWAN;
* IO (eng. Input/Output) - intrare/ieșire, interfața prin care dispozitivele comunică cu mediul lor exterior;
* HG - Hotărâre Guvernamentală;
* CSV (eng. Comma-Separated Values) - valori separate prin virgulă, un format de fișier utilizat pentru stocarea și schimbul de date tabelare;
* SQLite - O bibliotecă de programare care oferă o bază de date relațională ușoară și fără server;
* URL (eng. Uniform Resource Locator) - localizator uniform de resurse, adresa web care specifică locația unei resurse de pe internet;
* API (eng. Application Programming Interface) - interfață de programare a aplicației, un set de reguli și protocoale care permit diferitelor componente de software să comunice între ele;
* CI (eng. Continuous Integration) - integrare continuă, practica de a automatiza testarea și integrarea codului nou într-un proiect software în mod frecvent;
* CD (eng. Continuous Delivery) - livrare continuă, o practică în dezvoltarea software-ului care implică automatizarea procesului de livrare a software-ului la utilizatori;
* UML (eng. Unified Modeling Language) - limbajul de modelare unificată, un set de notări și reguli pentru a descrie și a modela structura și comportamentul sistemelor software;
* SysML (eng. Systems Modeling Language) - limbajul de modelare a sistemelor, o extensie a uml utilizată pentru modelarea și analiza sistemelor fizice și tehnice;
* AADL (eng. Architecture Analysis and Design Language) - limbajul de analiză și proiectare a arhitecturii, un limbaj de modelare utilizat în ingineria sistemelor embebuite și în timp real;
* OPM (eng. Object Process Methodology) - metodologia de proces orientată pe obiecte, o metodologie de dezvoltare a sistemelor care se concentrează pe modelarea obiectelor și a interacțiunilor dintre ele;
* RTC (eng. Real Time Counter/Clock) – contor în timp real;
* RAM (eng. Random Acces Memory) – memorie cu acces aleatoriu;
* Gnu (eng. GNU not Unix) – Gnu nu este Unix;
* ARM (eng. Advanced RISC (eng. Reduced Instruction Set Computer) Machine) – mașină RISC avansată.

INTRODUCERE

Aici se scrie o introducere despre ce este scris în memoriul explicativ ( raportul proiectului de licență)

În această lucrare vor fi prezentate rezultatele analizei domeniului de monitorizare a mediului. Rezultatele comparării a trei sisteme existente, Atmotube PRO, Garmin Tempe și DT-8820. Au fost definite scopul, obiectivele și specificațiile tehnice ale proiectului. A fost proiectat sistemul utilizând limbajul de modelare unificat UML.

Practica a avut loc în perioada 29 ianuarie – 9 februarie și 19 februarie – 1 martie. Conform contractului de angajare, practica a avut loc la compania Arobs Software S.R.L care pe piața din Republica Moldova este specializată în domeniul auto. În special programarea sistemelor încorporate din care sunt constituite autoturismele.

Ca activități practice pe lângă documentarea pentru realizarea acestui sistem a fost demarată și operațiunea de realizare a sistemului. La moment sistemul se află încă în etapa de dezvoltare, dar până la vara s-ar putea chiar să fie finalizat.

# ANALIZA DOMENIULUI DE STUDIU

În acest capitol se descrie domeniul din care face parte proiectul creat ( tehnologii informaționale). Ce tip de aplicație ( web, mobile...etc). În ce domeniu se va aplica proiectul (învățământ, sfera serviciilor......etc).

Aci se vorbește la general, nu se spune concret ce proiect voi face, insa se atrage atentai asupra domeniului, dar asta este prostesc

Monitorizarea mediului cuprinde metode și strategii pentru găsirea, analizarea și stabilirea parametrilor mediului. Aceasta poate duce la reglarea nivelurilor de poluare în cazurile când calitatea mediului nu este satisfăcătoare [1]. Factorii de bază pe care se concentrează această disciplină sunt:

* aerul;
* apa;
* solul;
* gălăgia;
* biodiversitatea.

Dintre factorii enumerați mai sus, aerul și gălăgia sunt centrul factorii de interes pentru această teză. Aceștia au un impact mare asupra sănătății umae și sunt relativ mai ușor de monitorizat de ceilalți factori.

Dintre tendințele tehnologice în domeniul de monitorizare al mediului cele mai relevante pentru tehnologiile informaționale sunt teledetecția și internetul lucrurilor. Teledetecția presupune utilizarea de sateliți, drone, avioane sau alte dispozitive pentru a capta imagini sau a efectua măsurători ale suprafeței sau atmosferei pământului. Internetul lucrurilor este rețeaua de obiecte fizice interconectate, cum ar fi senzori, dispozitive sau mașini, care pot comunica și schimba date [2]. Avantajul IoT față de teledetecție este că de obicei aceste dispozitive sunt mai ieftine și permit utilizarea lor pe o scară largă, totodată aceste dispozitive pot monitoriza în timp real mediul observat.

Internetul lucrurilor implementat pe scară largă permite colectarea unei cantități mai mare de date și luarea de decizii cu o precizie mai mare. Această tehnologie se axează în principal pe interacțiunea între dispozitive însă ca rezultat final se urmărește interacțiunea oeamenilor cu aceste dispozitive. Problemele principale cu care se confruntă acest domeniu sunt satisfacerea cerințelor non funcționale și anume asigurarea interoperabilității, scalabilității, fiabilității, securității, confidențialității și eticii dispozitivelor și sistemelor IoT. Interoperabilitatea se referă la capacitatea diferitelor dispozitive și sisteme de a comunica și de a lucra împreună fără probleme. Scalabilitatea se referă la capacitatea de a gestiona un număr tot mai mare de dispozitive și date fără a compromite performanța sau calitatea. Fiabilitatea se referă la capacitatea de a funcționa corect fără erori sau defecțiuni. Securitatea se referă la capacitatea de a proteja dispozitivele și datele împotriva accesului neautorizate sau atacurilor. Confidențialitatea se referă la capacitatea de a respecta drepturile și preferințele utilizatorilor cu privire la datele lor personale. Etica se referă la capacitatea de a adera la principiile și valorile morale atunci când proiectați și utilizați dispozitive și sisteme IoT [3].

Tehnologia informației este un set de domenii conexe care cuprind sisteme informatice; software; limbaje de programare; procesarea datelor și stocarea informațiilor [4]. Prin urmare IoT este un subdomeniul al tehnologiei informației având ca conceptele fundamentale:

* date;
* dispozitive;
* analiza;
* conectivitate.

## Importanța temei

Se va argumenta importanța creării unui astfel de proiect.

Ca consecință a evoluției civilizației umane și a tehnologiilor, oamenii au devenit o specie mai fragilă și mai vulnerabilă în fața mediului [5]. Deși depășirea acestor condinții este necesară pentru bunăstarea populației umane, monitorizarea mediului rămâne un aspect important în prevenirea deteriorării sănătății. Calitatea parametrilor mediului influențează direct sănătatea oricărui organizm viu. Spre exemplu, anual 9 milioane de oameni mor din cauza poluării [6]. Majoritatea țărilor nu depun eforturi considerabile pentru a rezolva această problemă [6]. Prin urmare este important ca măcar să conștientizăm nivelul de poluare la care suntem expuși.

Poluare nu poate fi evitată prin izolare de la lumea externă. Fiind că populația umană deja face acest lucru, populația globală petrece în jur de 90% în interior [7]. Analizând aceste informații, putem deduce că nivelul poluării în interior nu este mai mic ca nivelul de poluare în exterior, ba chiar poate fi mai mare așa cum morțile cauzate de poluare sunt în creștere [6] iar oamenii își petrec din ce în ce mai mult timp în interior.

O soluție evidentă, pentru a reduce efectele negative ale poluării asupra sănătății, este de a petrece mai mult timp în natură. Filosoful Friedrich Nietzsche a spus: “O viață sedentară este adevăratul păcat împotriva Duhului Sfânt.“. Acesta a ameliorat simptomele unei boli pe care le avea, făcând niște drumeții în natură.

Însă există posibilitatea ca mediul în care are loc odihna să fie poluat și prin urmare să nu aducă beneficii sănătății sănătății. Prin urmare monitorizarea mediului, este un instrument indispensabil pentru protejarea sănătății. Monitorizarea în timp real, cu feedback este și mai bună, așa cum nu este necesar de a investiga constant dacă starea mediului este satisfăcătoare. În caz că un parametru nu este corespunzător normelor, sistemul va înștiința utilizatorul.

## Sisteme similare cu proiectul realizat

Se va face o descriere a cel puțin 3 sisteme deja existente ( sau sisteme care au ceva comun din punct de vedere funcțional cu ce se va face în proiect).

Se va face o comparare a sistemelor descrise.

Pe piață există diferite sisteme de monitorizarea a mediului pentru diferite necesități. În această secțiune au fost identificate 3 sisteme cât mai portabile care să poată monitoriza parametrii care influențează sănătatea. Aceste 3 sisteme sunt:

* Atmotube PRO [8];
* Tempe [9];
* DT-8820 [10].

Calitățile sistemelor identificate care au fost analizate în detaliu, pentru a vedea care sunt punctele tari și slabe ale acestor sisteme sunt:

* numărul de parametrii măsurați;
* conectivitatea;
* durata de viață a bateriei;
* prețul.

Atmotube PRO, reprezentat în figura 1.1 este o stație meteo portabilă capabilă și să monitorizeze calitatea aerului. Aceasta poate monitoriza următorii parametri:

* particule PM1, PM2.5, PM10;
* compuși organici volatili (VOCs);
* temperatura aerului;
* umiditatea relativă;
* presiunea barometrică.

Funcționalități-le principale ale acestuia sunt monitorizarea parametrilor menționați anterior, această monitorizare are loc în timp real, iar datele sunt transferate pe telefonul mobil pentru a fi afișate prin intermediul Bluetooth.

Deși poate fi considerat portabil, dimensiunile acestuia pot fi considerate un dezavantaj, fiind relativ mari 86 x 50 x 22 mm. Ceea ce l-ar face incomod de purtat în mai multe situații.

Durata de viață a bateriei este de 10 zile. În comparație cu, celelalte sisteme identificate aceasta tot poate fi considerată un dezavantaj. Durata de viață a bateriei este relativ mică deoarece folosește ca tehnologie de comunicare Bluetooth.

Poate fi procurat cu 3400 de lei. Ceea ce îl face inaccesibil pentru un număr mare de persoane. Totuși prețul poate fi atribuit unui număr mai mare de parametri monitorizați, și faptului că dispozitivul poate să se conecteze la un telefon mobil.



Figura 1.1 - Atmotube PRO

Garmin Tempe, reprezentat în figura 1.2 este un mic senzor de temperatură, care se evidențiază prin dimensiunea sa mică de 18 x 12 x 18 mm. Dar nu oferă funcționalități suplimentare.

Funcționalitatea sa principală este monitorizare temperaturii. Afișarea are loc pe alt dispozitiv, care trebuie să fie compatibil cu protocolul ANT+.

El are două dezavantaje majore. Cum a fost menționat mai sus, datele pot fi afișate doar pe un dispozitive Garmin compatibil prin tehnologia ANT+. Al doilea dezavantaj este că poate monitoriza doar temperatura aerului.

Bateria acestuia are o durată de viață extraordinară de aproximativ un an. Asta se datorează faptului că dispozitivul are limitările menționate mai sus.

Acesta este cel mai accesibil sistem în această listă. Prețul acestuia este în jur de 800 de lei. Totuși considerând numărul de parametri măsurați, nu poate fi considerat un preț rezonabil.



Figura 1.2 - Garmin Tempe

DT-8820, reprezentat în figura 1.3 este un multimetru capabil să măsoare intensitatea luminii, intensitatea sunetului, temperatura aerului și umiditatea relativă.

Spre deosebire de celelalte dispozitive, DT-8820 dispune de un ecran pentru a afișa datele măsurate. Și chiar are și butoane fizice pentru a selecta modul al acestuia de lucru.

Dintre dezavantaje se numără lipsa conectivității, acesta poate fi utilizat ca un multimetru obișnuit, fără posibilitatea de a colecta datele. Dimensiunea acestui dispozitiv 251 x 64 x 40 mm, este un dezavantaj, deși poate fi considerat portabil, nu este foarte comod de a îl ține în buzunar.

Acest dispozitiv folosește baterii de 9V, și fiind ca nu este utilizat în continuu, asta ar presupune că durata de viață a unei baterii ar putea fi chiar mai mare de un an.

Prețul acestui dispozitiv este în jur de 1200 de lei. Pentru numărul de parametri care ii poate măsura pare a fi un preț mai rezonabil decât celelalte dispozitive.



Figura 1.3 - DT-8820

În urma comparării acestor dispozitive s-a stabilit că caracteristicile unui sistem dorit ar fi ca acesta să poată măsura câți mai mulți parametri posibil, dar cel puțin temperatura, umiditatea relativă, compușii organici volatili, nivelul de gălăgie și nivelul de iluminare. Ideal prețul acestui sistem ar trebui să fie cât mai accesibil. Cel puțin să nu depășească prețul de 600 de lei.

De la un astfel de sistem se urmărește în special ca dimensiunile acestuia să fie mici, cât mai aproape posibil de dimensiunile a Garmin Tempe de 18 x 12 x 18 mm. Și de asemenea cu o durată de viața a bateriei mare, de cel puțin 30 de zile. Dispozitivul trebuie să posede conectivitate prin Bluetooth pentru conectarea la un telefon mobil și char și suport pentru Wi-Fi pentru a transmite datele către un server MQTT.

## Analiza tehnologiilor

Un sistem IoT tipic are mai multe niveluri. Aceste niveluri depind unul de altul, importanța lor pentru sistem fiind în ordinea enumerării [11]:

* hardware dispozitiv;
* software-ul dispozitivului;
* protocoale de comunicare;
* platformă în nori.

La nivelul de hardware pe lângă mulțimea de senzori și mecanisme care vor fi utilizate, la baza dispozitivului se află un procesor. Opțiunile disponibile sunt: FPGA, Microcontroler sau ASIC [12]. Ce ține de dezvoltarea dispozitivului în caz că se optează pentru FPGA sau ASIC, va fi nevoie de scris cod în Verilog sau VHDL ceea ce este diferit de programarea clasică fiind că acest cod nu se execută secvențial pe un procesor dar este transformat în circuite care vor funcționa sincron sau asincron, după necesitate. Dacă se optează pentru microcontroler, atunci acesta poate fi programat cu limbaje precum C sau C++ dar și altele. Compararea acestor platforme hardware este prezentată în tabelul 1.1, fiecare tehnologie are avantajele și dezavantajele sale. Pentru un sistem care va fi produs pe scara mică, cel mai potrivit în majoritatea cazurilor este un microcontroler, așa cum acesta este mai ieftin de procurat în cantități mici, oferă suficientă flexibilitate în utilizarea acestuia însă este mai ușor de utilizat ca celelalte platforme.

Tabelul 1.1: Comparare FPGA, microcontroler, ASIC [12]

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteriu** | **FPGA** | **Microcontroler** | **ASIC** |
| **Cost (în cantități mici)** | moderat | mic | mare |
| **Cost (în cantități mari)** | moderat | mic | mic |
| **Viteza de procesare** | rapid | moderat | foarte rapid |
| **Consumul de energie** | moderat | mic | mic |
| **Flexibilitatea** | mare | mică | zero |
| **Ușurința de utilizare** | moderat | ușor | dificil |

Software-ul dispozitivului poate fi scris pentru a rula direct pe hardware, sau pentru a utiliza un sistem de operare, de cele mai multe ori, un sistem de operare în timp real [13]. Fiecare abordare are avantajele și dezavantajele sale. Avantajul scrierii codului fără OS este că performanța acestuia va fi mai mare. Avantajul utilizării unui RTOS este că acesta va garanta procesarea evenimentelor în timpul dorit. O funcționalitate importantă pe care o poate avea software-ul dispozitivului este programarea OTA, care permite actualizarea software-ului dispozitivului la distanță [11]. Ce ține de limbajele de programare care pot fi utilizate pentru a scrie acest software, nu există nici o limitare, pot fi utilizate atât limbaje compilate cât și limbaje interpretate. Din limbajele de programare compilate C este cel mai popular, estimându-se că acesta este utilizat în 80% din cazuri. C++ este utilizat din cauza că compilatorul suportă un standard mai nou de cât compilatorul de C. Dintre limbajele mai tinere care au început să fie populare și în domeniul sistemelor încorporate, însă de obicei nu în producție, sunt Rust și Zig [14]. Dintre limbajele interpretate, cele mai utilizate sunt Python și Lua.

Protocoalele de comunicare sunt aceleași ca și în cadrul altor sisteme. Pot fi identificate două categorii principale, protocoale de rețea și protocoale de date [15]. Dintre protocoalele de rețea cele mai utilizate sunt: Wi-Fi; Bluetooh; ZigBee; LoRaWAN. Dintre protocoalele de date cele mai utilizate sunt: AMQP; MQTT; HTTP; CoAP; DDS; LwM2M. La acest nivel nu este dificil de creat protocoale de date proprii.

Platforma în nori este utilizată pentru a prelua, prelucra, analiza și stoca datele. Printre platformele în norni populare se numără: Amazon Web Services; Microsoft Azure. Pe acest tip de platformă dezvoltatorii pot să ruleze aplicațiile lor proprii. Însă există și platforme în norni care oferă servicii specializate, cum ar fi platforma Adafruit IO care oferă servicii de broker MQTT.

## Parametrii mediului

Valorile de referință pentru parametrii mediului în diferite cazuri au fost extrase din HG353 din 2010 [16] și regulamentul sanitar privind normativele de emitere a zgomotului [17], însă acesta nu a fost încă aprobat. Aceste regulamente au ca scop asigurarea condițiilor de trai inofensive pentru populație și se axează pe parametrii în interior, unde oamenii îți petrec o mare parte a vieții. În continuare vor fi prezentate normele extrase din aceste surse în formă de tabel.

În tabelul 1.2 sunt prezentate limitele termice minime și maxime admise pentru diferite rate metabolice. Pentru a obține metabolismul în wați este nevoie de a converti cantitatea de Cal utilizată în Joule, acest lucru se face înmulțind Cal cu 4184, apoi pentru a obține wați este nevoie de împărțit energia la numărul de secunde în care au fost consumate caloriile [18]. De asemenea din acest regulament mai poate fi evidențiat faptul că în perioada rece a anului, se recomandă ca temperatura în interior să fie cu, cel puțin 3 °C mai mare ca în perioada caldă.

Tabelul 1.2 – Limitele termice minime și maxime admise la posturile de lucru [16]

|  |  |  |
| --- | --- | --- |
| ****Metabolismul,** (M) W** | ****Temperatura aerului minimă,** °C** | ****Temperatura aerului maximă,** °C** |
| M ≤ 117 | 18 | 32 |
| 117 < M ≤ 234 | 16 | 29 |
| 234 < M ≤ 360 | 15 | 26 |
| 360 < M ≤ 468 | 12 | 22 |
| M > 468 | 12 | 18 |

Indiferent de anotimp se recomandă ca umiditatea relativă în încăperi să fie menținută între 30 % și 70 % [16]. Umiditatea ridicată amplifică sensibilitatea la temperaturi reci, de aceia se recomandă ca dacă umiditatea aerului este mare, temperatura aerului să menținută la o valoarea mai ridicată.

În tabelul 1.3 sunt prezentate nivelurile de iluminare recomandate, pentru lucrărilor cu diferite caracteristici. Aceste recomandări sunt potrivite atunci când contrastul între detaliul observat și fond este mediu și când fondul nu este nici luminos nici întunecat. În timpul somnului se recomandă să nu fie lumină, aceasta poate cauza probleme cu vederea pe termen lung

Tabelul 1.3 – Nivelul de iluminare recomandat, pentru contrastul mediu între detaliu și fond [16]

|  |  |
| --- | --- |
| **Caracteristicile lucrărilor vizuale, pentru distanța de privire de 344 mm** | **Nivelul normat de iluminare, lx** |
| Lucrări de precizie deosebită cu detalii sub 0,1 mm | 1500 |
| Lucrări de precizie foarte mare cu detalii detalii între 0,1 mm şi 0,3 mm | 750 |
| Lucrări de precizie mare cu detalii între 0,3 mm şi 0,5 mm | 500 |
| Lucrări de precizie medie cu detalii între 0,5 mm şi 0,8 mm | 300 |
| Lucrări de precizie mică cu detalii între 0,8 mm şi 1,2 mm | 200 |

În tabelul 1.4 sunt prezentate nivelurile sonore maxime admisibile în încăperile locative, conform legii, după ora 22 este nevoie ca zgomotele emise să nu depășească 45 dBA. Aceste date sunt caracteristice pentru apartamente, pentru cămine spre exemplu este permis ca nivelul zgomot să fie mai mare cu 5 dBA în ambele cazuri.

**Tabelul** **1.4 – Nivelurile sonore maxime ale zgomotului admise în încăperile locative** **[17]**

|  |  |
| --- | --- |
| **Intervalul timpului de evaluare** | **Nivelurile sonore maxime, dBA** |
| 7.00-22.00 | 55 |
| 22.00-7.00 | 45 |

# SCOPUL, OBIECTIVELE ȘI CERINȚELE SISTEMULUI

Problema întâmpinată în ceea ce privește monitorizarea mediului constă în faptul că majoritatea sistemelor existente pentru măsurarea unei varietăți mari de parametri mai degrabă sunt niște multimetre decât niște dispozitive portabile, care să funcționeze autonom și doar să înștiințeze utilizatorii în caz de pericol pentru sănătate. Deși există sisteme care ar face parțial acest lucru, ele oferă un număr limitat de parametri pe care îi pot monitoriza, sau nu oferă suficiente funcționalități.

Prin dezvoltarea unui sistem portabil pentru monitorizarea factorilor mediului se urmărește scopul de a identifica dacă mediul din proximitatea utilizatorului prezintă un risc pentru sănătatea acestuia. După cum a fost stabilit în capitolul precedent, temperatura, umiditatea, gălăgia, iluminarea dar și alți parametri poate afecta sănătatea umană, de aceea este recomandat de a îi menține în normele stabilite. Pe lângă normele identificate există și alte standarde ce definesc valorile optime pentru majoritatea parametrilor, în special cei care pot fi controlați în interior. Este râvnită monitorizarea unui număr cât mai mare de parametri ai mediului. Însoțit de o portabilitate ridicată și de o autonomie mare a bateriei. Însă din cauza constrângerilor de resurse materiale, dimensiuni ale viitorului sistem și autonomiei acestuia se va opta ca acest sistem să poată monitoriza cel puțin:

* temperatura;
* compușii volatili organici;
* nivelul de iluminație;
* nivelul de gălăgie.

Pentru a realiza acest proiect se propun următoarele obiective:

* analiza și validarea cerințelor sistemului;
* dezvoltarea dispozitivului de monitorizare a mediului;
* monitorizarea parametrilor doriți;
* asigurarea portabilității dispozitivului;
* dezvoltarea aplicației mobile;
* stabilirea conexiunii între dispozitiv și aplicația mobilă;
* implementarea alarmelor;
* crearea unei interfețe de utilizator minimalistă;
* monitorizarea mediului în timp real;
* minimizarea costurilor de producție;
* testarea sistemului în condiții reale;
* documentarea și raportarea rezultatelor.

După cum a fost stabilit sistemul de interes este constituit din două elemente, dispozitivul și aplicația mobilă. Aceste elemente la rândul lor pot fi considerate sisteme, așa cum se dorește ca ele să poată funcționa independent unul față de altul într-o oarecare măsură. Un aspect important al sistemului este autonomia față de conexiunea la internet, sistemul nu trebuie să depindă de aceasta pentru a funcționa, această problemă a fost identificată în cadrul aplicației mobile de la Garmin. În subcapitolele ce urmează, vor fi expuse cerințele funcționale ce descriu comportamentul și cerințele non funcționale ce descriu calitățile, pentru sistemul în ansamblu. Cerințele pentru fiecare element în parte pot fi identificate din aceste cerințe, însă specificarea lor în parte nu oferă multe beneficii proiectului, așa cum acesta nu este foarte complex.

## Cerințe funcționale

* CF1: Sistemul trebuie să fie capabil să monitorizeze temperatura aerului, umiditatea relativă a aerului, compușii volatili organici, intensitatea luminii și nivelul de zgomot. Sistemul poate măsura și alți parametri dacă senzorii utilizați pot măsura și alți parametri. Pe lângă acești parametri trebuie să se monitorizeze și nivelul bateriei.
* CF2: Eșantionarea datelor trebuie să respecte o recurență predefinită, implicită de 1 minută dar configurabilă, pentru fiecare parametru în parte. Configurarea recurenței trebuie să poată fi efectuată în timpul operației sistemului. Rata de eșantionare configurată trebuie să fie validată, având ca limite admisibile, minimă de 30 secunde și maximă de 24 de ore. Sistemul trebuie să permită dezactivarea monitorizării parametrilor nedoriți.
* CF3: Sistemul trebuie să asigure persistența datelor. Persistența datelor trebuie să fie asigurată local, fără servicii de la al furnizori terți. Baza de date utilizată trebuie să fie SQLite. Sistemul trebuie să ofere posibilitatea de a exporta datele, fie în CSV sau altele.
* CF4: Sistemul trebuie să ofere posibilitatea de a seta alarme. În figura 2.2 este prezentată interfața de utilizator care permite configurarea alarmei selectate. Interfața de utilizator trebuie să permită selectare unui interval, iar alarma va fie activată dacă valoarea actuală este în interiorul sau exteriorul intervalului în dependență de setare. În caz că există mai multe dispozitive înregistrate în sistem, o alarmă poate fi activată pentru mai multe dispozitive.
* CF5: Sistemul trebuie să ofere posibilitatea de a vizualiza a datelor prin mai multe reprezentări grafice. Este necesară reprezentarea fiecărui parametru într-o listă sau grilă, și de-asemenea de prezentat grafice evoluția datelor în timp.
* CF6: Componentele sistemului adică dispozitivul și aplicația mobilă trebuie să comunice printr-un protocol propriu de transfer de date, care să funcționeze la un nivel mai sus de protocolul de rețea Bluetooth. Interacțiunea cu dispozitivul va fi realizată prin intermediul aplicației mobile.
* CF7: Sistemul trebuie să fie inter-operabil cu alte sisteme existente. Pentru realizarea acestei funcționalități sistemul va trebui să ofere o interfață compatibilă cu protocolul MQTT. În configurarea conexiunii, vor fi prezente câmpuri precum URL-ul, cheia API, numele de utilizator și parola. Aplicația trebuie să valideze conexiunea și să nu necesite toate câmpurile în cazul când unele nu sunt necesare.
* CF8: Interfața de utilizator trebuie să fie minimalistă și să nu distragă utilizatorii din scopurile lor pentru a utiliza sistemul. În figurile 2.1 și 2.2 este prezentat un prototip inițial al paginilor aplicației mobile, realizat în Figma. La lansarea aplicației, va fi afișată o pagină de pornire pe care este scrisă denumirea aplicației mobile. Apoi, urmează pagina de pornire, care va varia în funcție numărul de dispozitive conectate la aplicația mobilă. În cazul în care există mai multe dispozitive conectate, aplicația va prezenta o pagină de pornire cu o vizualizare succintă pentru toate dispozitivele conectate, oferind astfel o imagine de ansamblu a stării acestora. La apăsarea pe un parametru, trebuie să fie afișat un grafic liniar cu evoluția acestuia în timp. Pe pagina principală, utilizatorii vor avea posibilitatea de a conecta noi dispozitive, facilitând astfel extinderea rețelei de monitorizare. De asemenea, în partea de sus stângă a interfeței va există butonul de setări care deschide pagina de setări, unde vor putea fi adăugate conexiuni MQTT, și va putea fi configurată recurența eșantionării. Pe pagina principală în partea dreaptă sus va fi un buton pentru alarme, care va deschide pagina de alarme, în care vor putea fi create, activate, dezactivate, editate și șterse alarmele.

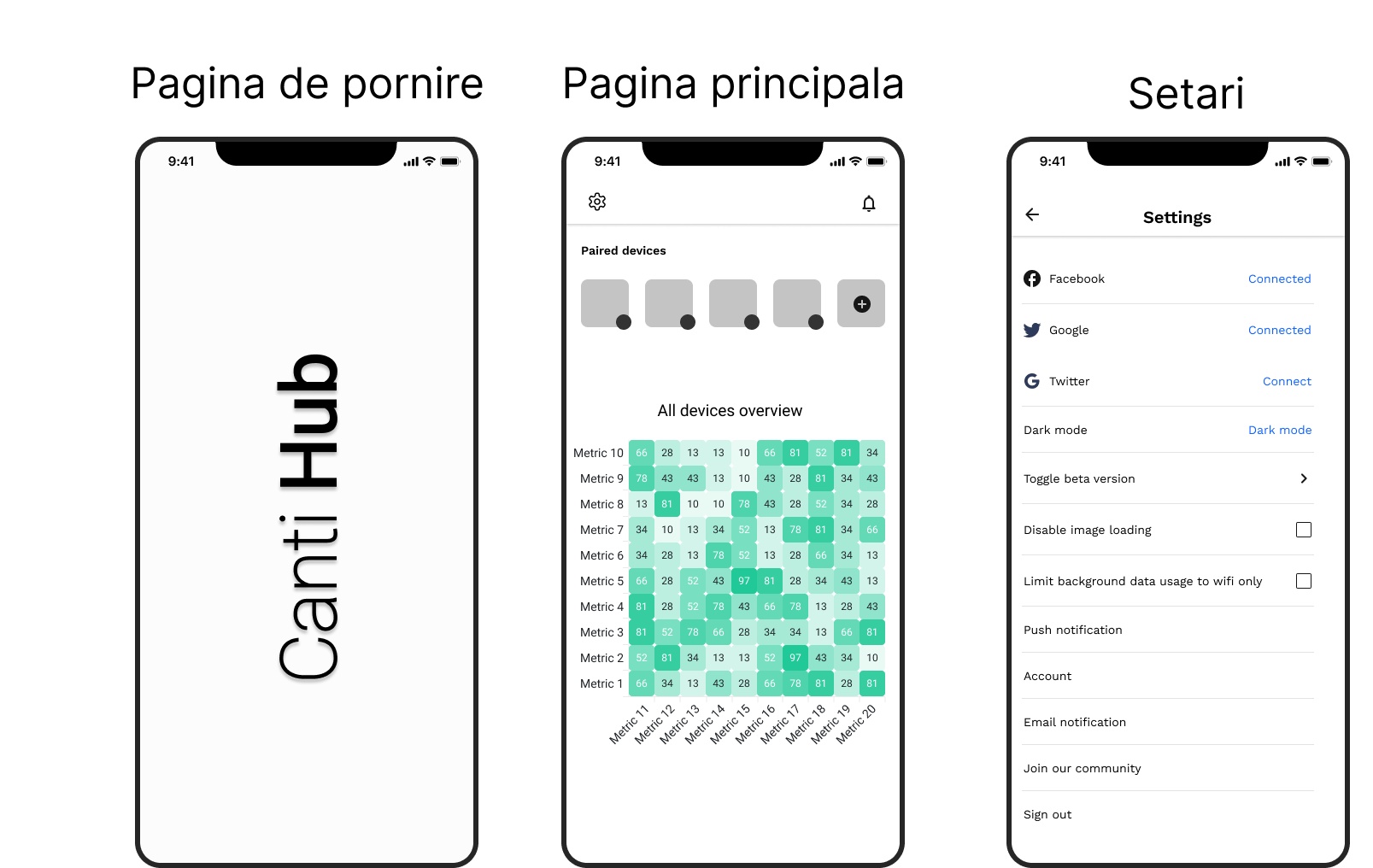


Figura 2.1 - modelul paginii de pornire, principale și setări

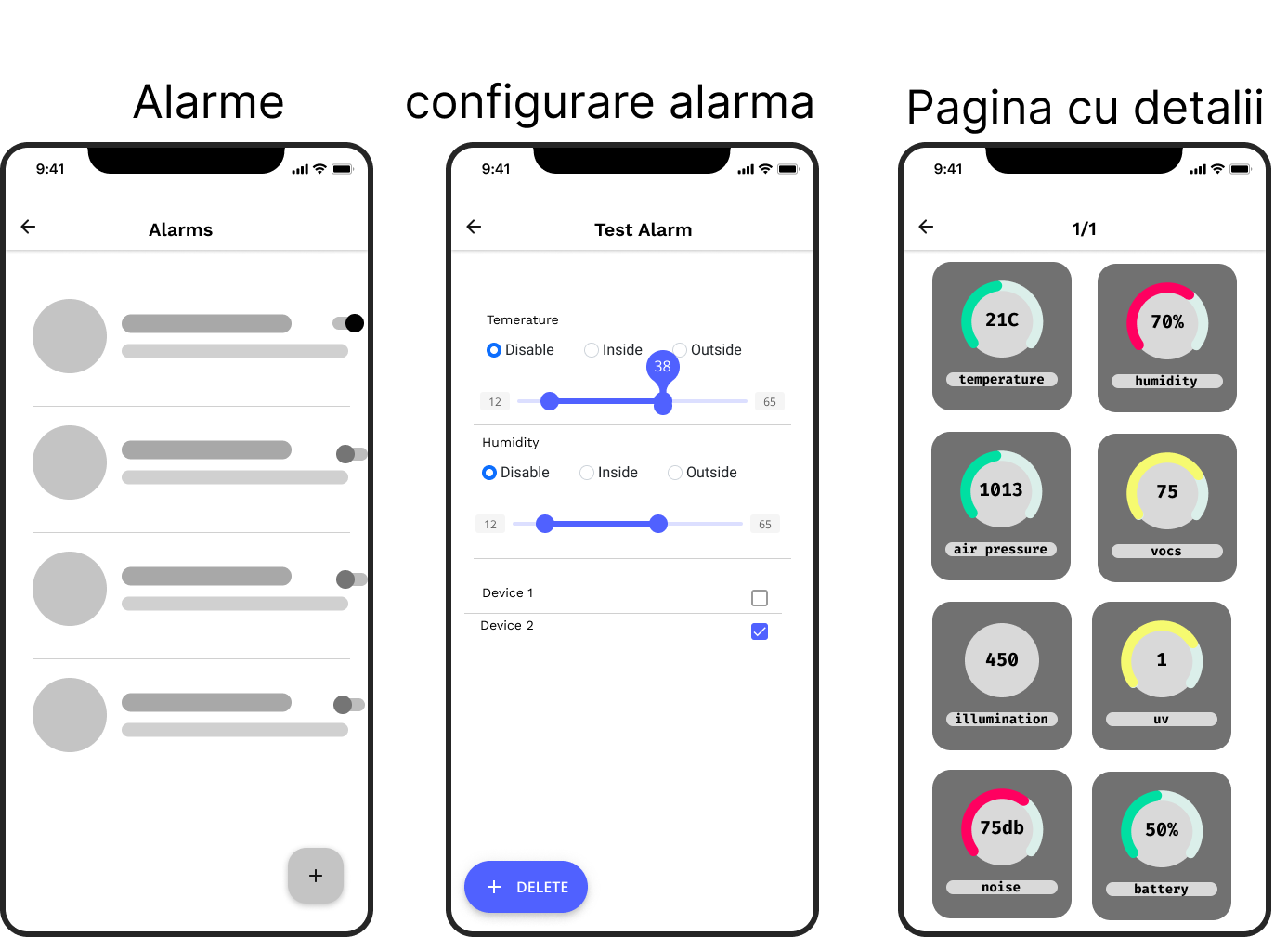


Figura 2.2 - modelul paginii pentru alarme, configurări și detalii

## Cerințe non-funcționale

* CNF1: Sistemul trebuie să aibă o autonomie ridicată. În primul rând, funcționarea sistemului nu trebuie să depindă de conexiunea la internet. În al doilea rând, părțile componente ale sistemului nu trebuie să depindă de conexiunea Bluetooth pentru a funcționa. Sistemul trebuie să aibă o durată de viață a bateriei de 30 de zile fără a necesita reîncărcarea dispozitivului. În plus, sistemul trebuie să poată detecta și trata potențialele erori pentru a rămâne funcțional.
* CNF2: Sistemul trebuie să fie scalabil, în sensul că la o aplicație mobilă să se conecteze mai multe dispozitive.
* CNF3: Interoperabilitatea va fi asigurată, prin implementarea unei interfețe MQTT pentru a posta sau prelua date de la sisteme externe. Pe lângă asta se propune opțiunea de a avea dispozitive virtuale care nu sunt conectate prin Bluetooth dar datele colectate de ele sunt afișate în aplicație.
* CNF4: Asigurarea unui nivel minim de securitate este esențială. Conectarea la dispozitiv trebuie să fie realizată printr-un mecanism de autorizare.
* CNF5: Sistemul trebuie să fie adaptabil. Trebuie să poată suporta diferite versiuni de hardware și software pentru dispozitivele conectate, iar acestea trebuie identificate și gestionate corespunzător.
* CNF6: Codul trebuie să fie cât mai portabil pentru a putea fi ușor adaptat și portat pe alte platforme, în cazul în care este necesar. Este necesar ca software-ul să fie scris într-un limbaj compilat, este de dorit utilizarea limbajului Zig.
* CNF7: Prețul de producție al unui dispozitiv trebuie să fie minimizat, dar totodată să se țină cont de numărul de parametri pe care acesta îi poate măsura.
* CNF8: Dimensiunile dispozitivului nu trebuie să depășească dimensiunile de 86 x 50 x 22 mm.
* CNF9: Realizarea sistemului trebuie să fie versionată cu ajutorul instrumentului Git. Și să se respecte convenția de mesaje **conventional commit**.
* CNF10: Sursele sistemului trebuie să fie găzduite pe platforma Github, după posibilitate să fie implementate pipeline-uri pentru integrare continuă și livrare continuă. Pentru a facilita procesul de CI și CD pot fi utilizate containere docker.
* CNF11: Realizarea sistemului va avea loc după modelul de dezvoltare iterativ – agile. Acest lucru se datorează faptului că există mai multe momente în cadrul dezvoltării proiectului, când va trebui să se pună accentul pe documentare iar dezvoltarea să fie oprită.
* CNF12: Sistemul trebuie să fie simplu, să nu fie supra-proiectat. Funcționalități mai avansate pot fi adăugate după ce sistemul este într-o stare funcțională.
* CNF13: Sistemul trebuie să fie extensibil, trebuie să permită adăugarea de noi funcționalități cu ușurință.

# MODELAREA ȘI PROIECTAREA SISTEMUL

Se va face o mică introducere despre proiectarea unui sistem.

Se va descrie:

* ce metodă de proiectare s-a ales;
* ce limbaj de proiectare s-a ales (UML);
* ce instrument (mediul de dezvolatre) s-a utilizat.

Pentru modelarea sistemului, au fost considerate mai multe limbaje de modelare. Acestea fiind: UML, SysML, Archimate, AADL și OPM. La prima vedere AADL [19] părea a fi o alegere mai bună, considerând că sistemul va avea și partea hardware. Acesta ar fi permis abordare de inginerie a sistemelor bazată pe modele, care permite efectuarea simulărilor și identificarea posibilelor erori înainte de realizarea sistemului. Totodată acest limbaj oferă atât limbaj textual cât și grafic. Însă din cauza că pe internet este un număr restrâns de resurse pentru a învăța nu s-a optat pentru a îl utiliza. A doua ce-a mai bună abordare ar fi de a modela sistemul pe foi de hârtie utilizând un pix, așa cum mediul digital nu se poate compara cu mediul analog, însă această abordare nu ar fi acceptabilă în ipostaza proiectului respectiv. Ca urmare a acestor fapte s-a optat pentru utilizarea limbajului UML [20] – limbaj de modelare unificat. Deși inițial acesta era prevăzut doar pentru sistemele intensive de software, în următoarele reviziuni a fost introdusă posibilitatea de a modela sisteme complete. Deși unii încă au rămas nemulțumiți și au creat limbajul derivat SysML care se concentrează pe sisteme dar nu software.

Ca instrument pentru modelare sistemului a fost ales Modelio [21], acesta a fost ales datorită faptului că este gratis, cu, codul sursă deschis. Se instalează și rulează pe mașina locală fără a avea nevoie de acces la internet. Deși acest instrument nu este perfect și are probleme cu reprezentarea unor diagrame, faptul că oferă o experiență de utilizare intuitivă îi oferă un avantaj față de soluțiile comerciale.

În specificațiile v2.5.1 de UML [20] sunt definite 14 diagrame [22], care sunt împărțite în două tipuri, diagrame comportamentale și diagrame structurale. Aceste diagrame sunt prevăzute pentru diverse cazuri și utilizarea tuturor poate fi justificată doar pentru cele mai complexe proiecte. De aceia pentru proiectarea sistemului au fost utilizate doar 8 tipuri de diagrame. Aceste diagrame sunt:

* diagrama cazurilor de utilizare;
* diagrama de activitate;
* diagrama de secvență;
* diagrama mașină de stări;
* diagramă de comunicare;
* diagramă de clase;
* diagramă de componente;
* diagramă de implementare.

Informația ce-a mai relevantă despre sistem este în diagrama de amplasare, așa cum acest sistem are și o parte hardware. Din această poate fi dedus faptul că topologia sistemului va fi stea. Însă din cauza limitărilor limbajului UML, acest lucru ar putea fi sărit cu vederea.

Deși modelarea are loc înainte de realizare, în timpul dezvoltării pot fi identificate probleme în model, sau limitații tehnice. Ca urmare modelul va suferi schimbări, pentru a ilustra structura și comportamentul sistemului în realitate.

## Descrierea comportamentală a sistemului

Se va face o introducere ce prezintă descrierea comportamentală și pe scurt ce include:

* imaginea generală asupra sistemului;
* modelarea vizuală a fluxurilor;
* stările de tranzacție a sistemului (Statechart Diagram, minim 3 diagrame);
* descrierea scenariilor de utilizare a aplicaţiei (Sequence Diagram, minim 3 diagrame);
* fluxurile de mesaje şi legăturile dintre componentele sistemului (Collaboration Diagram, minim 2 diagrame).

**P.S: numărul de diagrame depinde de sistemul proiectat, dacă sistemul permite atunci vă conduceți de recomnadări.**

Așa cum a fost menționat, limbajul UML are un set de diagrame care descriu comportamentul sistemului. Aceste diagrame sunt:

* diagrama cazurilor de utilizare;
* diagrama de activitate;
* diagrama de secvență;
* diagrama mașină de stări;
* diagramă de comunicare;
* diagramă de interacțiune;
* diagramă de sincronizare.

Diagrama cazurilor de utilizare, descrie ce face un sistem, însă nu descrie modul de implementare. În această diagramă este prezentat un set de evenimente care pot fi declanșate de către actor. Actorul în această diagramă poate fi o persoană, un rol, sau chiar și un sistem. Prin această diagramă poate fi prezentată imaginea generală a sistemului.

Diagrama de activitate prezintă fiecare pas pentru a atinge un obiectiv în cadrul unui sistem. Această diagramă este potrivită pentru reprezentarea algoritmilor. Fiecare activitate are un punct de intrare și un punct de ieșire. Iar fiecare acțiune din interior, poate fi la rândul ei un proces.

Diagrama de secvență este potrivită pentru a arăta interacțiunea dintre mai multe obiecte sau actori. Aceasta are linii de viață din care sunt trase mesaje către alte linii. Această diagramă prezintă în mod cronologic mesajele care au avut loc între obiecte și actori.

Diagrama mașină de stări este utilizată pentru a modela comportamentul unui obiect complex. Aceasta prezintă care sunt evenimentele care pot schimba starea obiectului și ce acțiuni sunt întreprinse când o stare este activă.

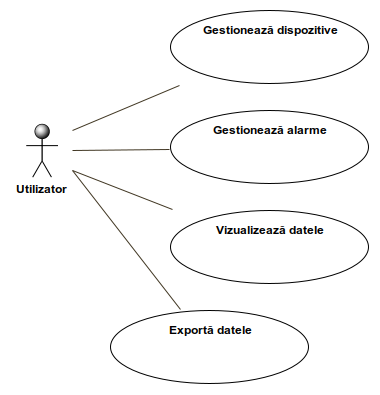
Diagrama de comunicare pune accent pe comunicarea dintre obiecte. Aici sunt trase asocieri între obiecte, iar mesajele și direcțiile acestora sun puse lângă linie. Conține aceiași informație ca o diagramă de secvență.

Diagrama de interacțiune adesea este complexă și este evitat de nespecialiști și amatori. Aceasta pe lângă elementele diagramei de activitate conține elemente cum ar fi interacțiunea, constrângeri de timp sau de durată.

Diagrama de sincronizare cunoscută și cu numele de diagramă de evenimente prezintă cronologia în are obiectele și actorii acționează. Se pune accentul pe durata evenimentelor și modificările care se produc. Însă din cauza că Modelio nu le suportă. Această diagramă nu a fost utilizată în modelarea sistemului.

### Imaginea generală asupra sistemului

Pentru a utiliza sistemul, utilizatorul trebuie să dețină cel puțin un dispozitiv. Apoi acesta va trebui să îl înregistreze în sistem. Odată ce dispozitivul este înregistrat în sistem utilizatorul poate vizualiza datele monitorizate pe acesta. Poate export datele respective. De asemenea mai există și posibilitatea de a seta alarme. Aceasta este imaginea generală a sistemului, reprezentarea ei grafică poate fi vizualizată în figura 3.1.

Figura 3.1 - Utilizarea sistemului

Sistemul va permite vizualizarea datelor în mai multe moduri. Cel mai de bază mod va fi fie o listă fie o grilă cu fiecare parametru monitorizat, va fi o pagină dedicată fiecărui dispozitiv. Altă reprezentare va fi o diagramă de tip arie, normalizată astfel în cât să poată fi identificat cu ușurință dacă un parametru este prea mare sau prea mic. În ultimul rând va fi prezentă posibilitatea de a vizualiza fiecare parametru în parte pe un grafic cu linii, pentru a putea vedea cum a evoluat acest parametru. Reprezentarea acestor cazuri de utilizare poate fi vizualizată în figura 3.2.

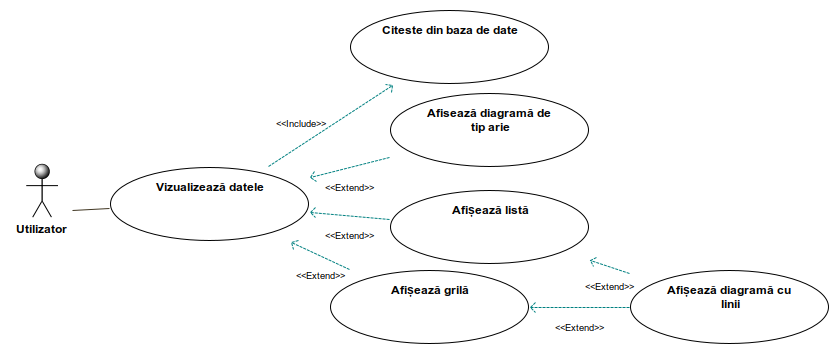


Figura 3.2 - Vizualizarea datelor

În figura 3.3 este prezentat mai detaliat modul de gestionare al sistemului. În primul rând, așa cum a fost menționat anterior, utilizatorul poate adăuga dispozitive în sistem. Totodată utilizatorul le poate șterge. Un alt aspect important al gestionării sistemului este configurarea recurenței. Utilizatorul poate configura recurența pentru fiecare parametru în parte. Respectând condiția că aceasta să se încadreze în intervalul 29 secunde, 24 ore. În cadrul acestui sistem, utilizatorul poate gestiona alarme, asta presupune crearea, ștergerea, dezactiva și activa alarmele. De asemenea el le poate edita. Toate aceste setări sunt transmise și către dispozitiv dacă bluetoohul este activat și dispozitivul conectat este în raza de acțiune.

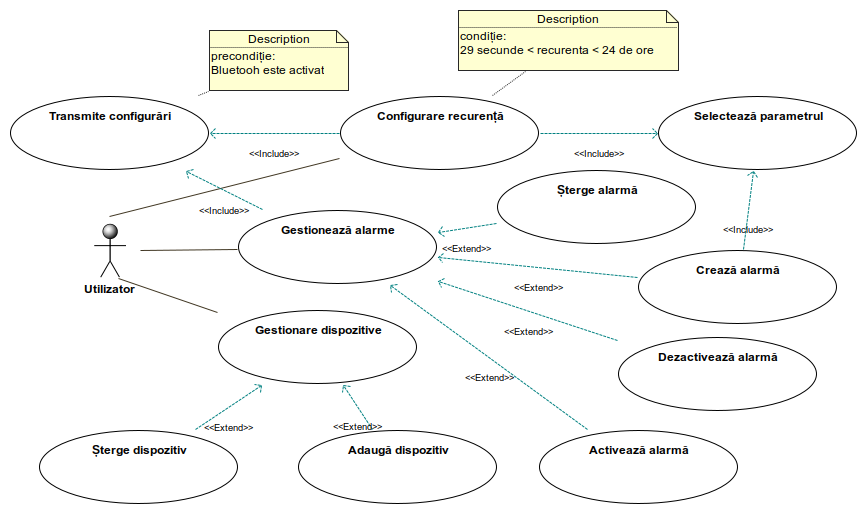


Figura 3.3 - Gestionarea sistemului

Inspectând figura 3.4, poate fi dedus că sistemul este unul inter-operabil. Există posibilitatea de a conecta sistemul la o platformă externă. Acest lucru va avea loc prin intermediul protocolului de comunicare MQTT. Aplicația va fi principalul publicator către această platformă externă. Pentru ca dispozitivul să poată transmite datele prin MQTT, acesta va trebui să fie în regim staționar. Regimul staționar presupune că dispozitivul este conectat la o sursă de alimentare.

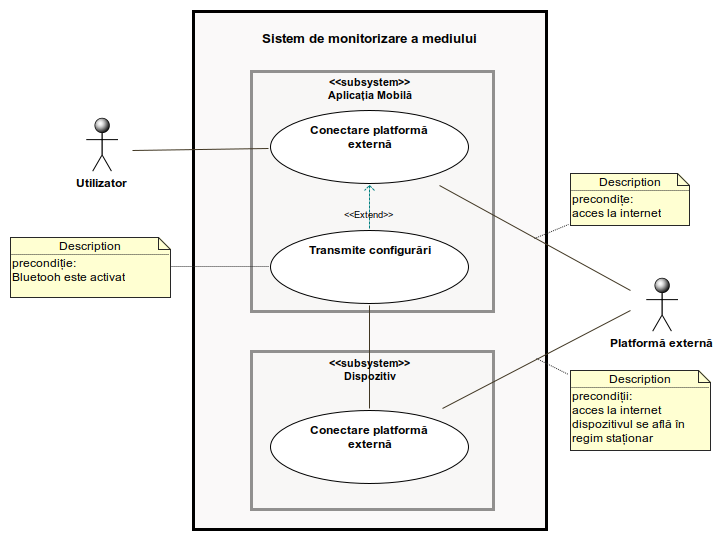


Figura 3.4 - Conectarea la o platformă externă

În figura 3.5 este prezentat modul în care software-ul sistemului poate fi actualizat. În primul rând dezvoltatorul trebuie să publice o versiune nouă de software și să o marcheze corespunzător. Apoi GitHub va compila software-ul fie că acesta este pentru dispozitiv sau telefon mobil. Odată ce utilizatorul va verifica dacă există o nouă versiune software, aplicația îl va înștiința de acest lucru. Versiunea nouă de software va fi descărcată. Utilizatorul va trebui să instaleze manual noua versiune de aplicație mobilă, fiind că pe dispozitivele care nu sunt rutate, instalarea aplicațiilor în fundal este posibilă doar prin intermediul serviciilor google. Noua versiune de software pentru dispozitiv va fi transmisă prin Bluetooth către dispozitiv. Dacă acesta avea cel puțin 70% de baterie, actualizarea va avea loc. Dacă nu utilizatorul va fi înștiințat de acest lucru. Și îi va fi propus să pună dispozitivul la încărcat.

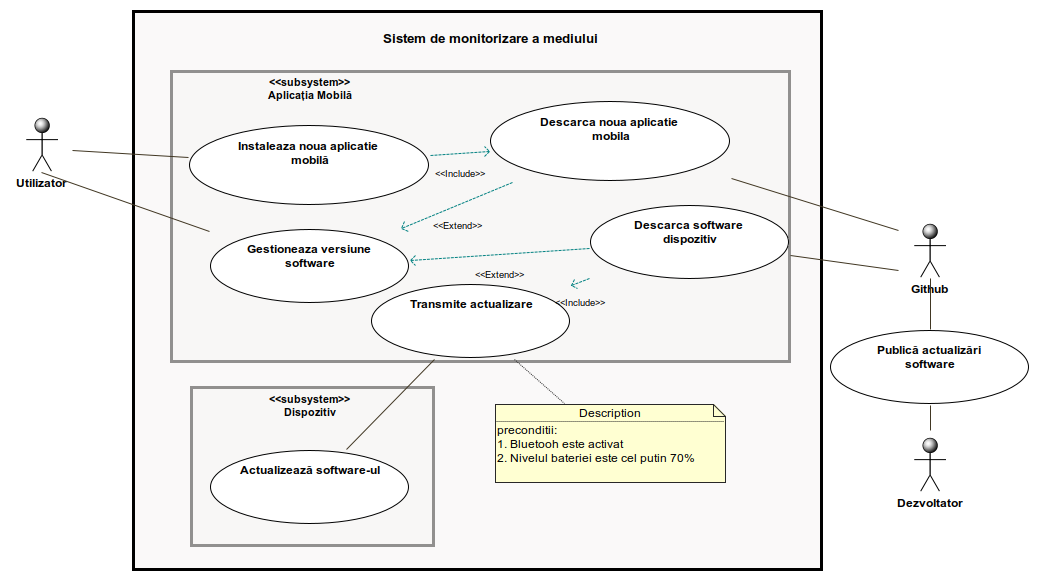


Figura 3.5 - Actualizarea versiunii software

### Modelarea vizuală a fluxurilor

Fiecare senzor cu, care este dotat dispozitivul va fi utilizat pentru a citi unul sau mai mulți parametri. Primul pas este să fie inițializat senzorul. Aici există posibilitatea ca senzorul să fie defect, ca rezultat nu se va primi nici un răspuns, acest caz va trebui tratat ca o eroare. În cazul când senzorul este funcțional, se începe măsurarea parametrului. Se așteaptă finisarea operației. Însă în acest timp pot fi efectuate și alte operațiuni. Spre exemplu pot fi inițializați și alți senzori. Odată ce parametrul a fost convertit, software-ul îl poate citi. În figura 3.6 este reprezentat acest proces.

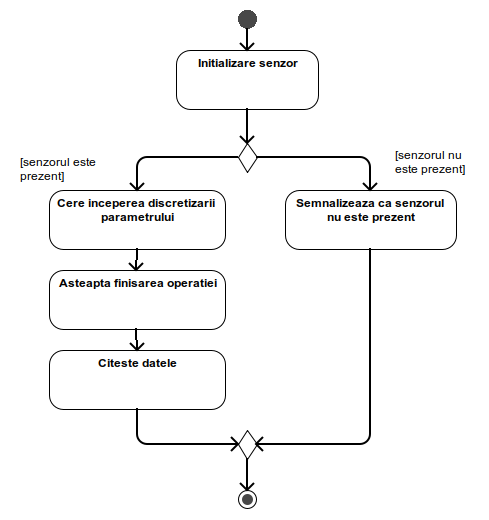


Figura 3.6 - Citirea parametrului

În figura 3.7 sunt prezentați pașii pentru a evalua alarmele. În primul rând sunt determinate care alarme sunt activate, pentru a nu genera alarme false. Asta este pe partea de aplicație mobilă, fiind că pe partea de dispozitiv va fi permis de setat doar 3 alarme, pentru a nu utiliza prea multă memorie. Următorul pas este de a identifica dacă condițiile pentru măcar o alarmă sunt îndeplinite. Condiția prezintă un interval în care valoarea parametrului curent trebuie să se încadreze sau să nu se încadreze pentru a genera alarma. Acest comportament este specificat de utilizator. Apoi dacă există măcar o alarmă care să fie activă, va fi generat un semnal de alarmă. Pe aplicația mobilă poate fi prezentată numele alarmei.

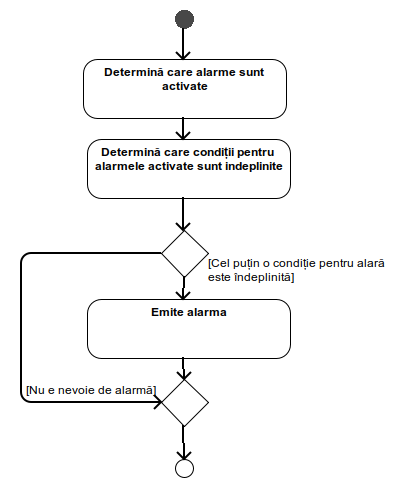


Figura 3.7 - Evaluare alarme

Exportarea datelor este un proces simplu. În primul rând datele sunt extrase din baza de date. Apoi în dependență de formatul selectat de către utilizator, datele sunt transformate într-un fișier de acest format. Formatele necesare de implementat sunt CSV și SQLite, însă se poate de implementat și alte formate. În ultimul rând utilizatorul este întrebat unde dorește să salveze fișierul. Acest proces poate fi vizualizat în figura 3.8.

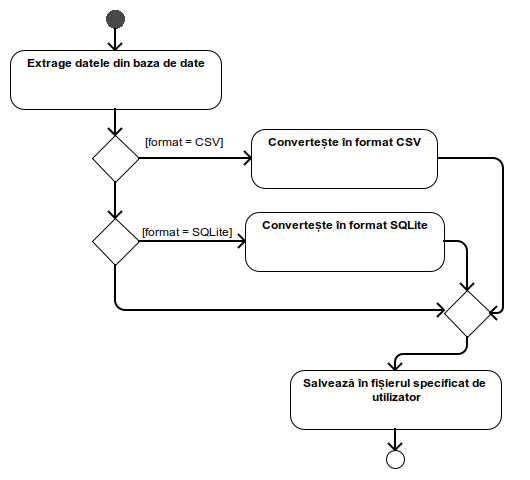


Figura 3.8 - Exportarea datelor

### Stările de tranzacție a sistemului

Dispozitivul se poate afla în patru stări principale. Stările sunt Inițializare, AutoTestare, Operație și Somn. Dispozitivul ajunge în cadrul stării de Inițializare odată ce acesta este pornit. Pornirea este fizică, prin conectarea alimentării. În cadrul acestei stări microcontrolerul va inițializa periferialele sale și senzorii. Dacă inițializarea va avea loc cu succes dispozitivul va trece în starea de AutoTestare, dacă nu acesta se va stinge. Când se află în starea de AutoTestare microcontrolerul va testa periferialele sale, integritatea senzorilor și a altor componente. În caz că testarea rezultă în erori dispozitivul se va duce înapoi în sarea de Inițializare. Însă dacă mai mult de trei cicluri au fost cu eroare, dispozitivul se va stinge. Dacă testarea s-a finisat cu succes, atunci dispozitivul trece în starea de Operațiune în cadrul căreia el va colecta datele și le va transmite după posibilitate. Totodată pentru a reduce din consumul de energie, așa cum de regulă citirea unui parametru durează câteva milisecunde iar restul, cel puțin 29 de secunde nu va fi nimic de făcut, dispozitivul se va duce în starea de Somn. În aceasta majoritatea periferialelor dispozitivului sunt deconectate. Apoi după ce a expirat timpul de inactivitate, dispozitivul se întoarce în starea de Operațiune. În figura 3.9 pot fi vizualizate aceste stări în mai puține cuvinte.

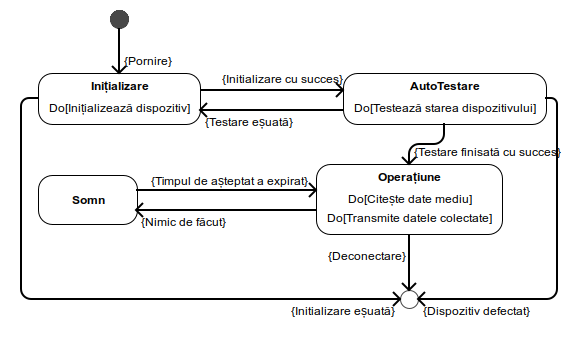


Figura 3.9 - Stările dispozitivului

În figura 3.10 sunt prezentate stările parcurse atunci când dispozitivul se află în starea de Operațiune. În primul rând în starea de Configurare se începe măsurarea datelor pe senzori. Apoi se trece în starea de așteptare, în care se așteaptă finisarea convertirilor. După ce convertirea a fost efectuată, datele pot fi citite. Apoi se trece la starea de Procesare, asta ar presupune filtrarea acestora. În caz că este posibil datele sunt transmise. Apoi dispozitivul va trece în starea de Somn.

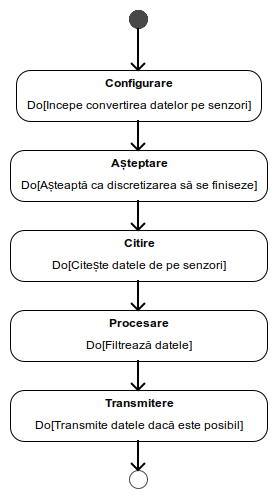


Figura 3.10 - Sub-stările etapei de operațiune

### Descrierea scenariilor de utilizare a aplicației

Dispozitivul este un element autonom al sistemului. Odată ce acesta este alimentat, el inițiază senzorii și apoi începe procesul de colectare al datelor. Acest proces repetitiv presupune inițializarea măsurării parametrilor, citirea parametrilor și salvarea acestora în caz că ei nu pot fi transmiși. Când utilizatorul are nevoie de aceste date, el va accesa aplicația mobilă. Aceasta la rândul ei va obține datele de la dispozitiv și le va afișa pentru utilizator. Acest scenariu poate fi vizualizat în figura 3.11.

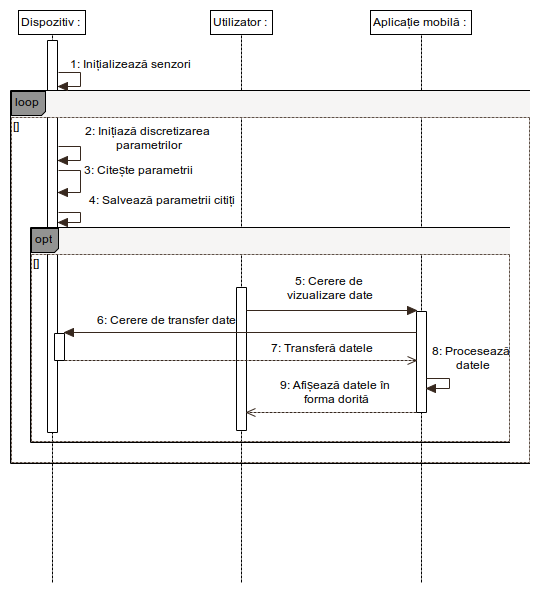


Figura 3.11 - Colectarea datelor

În figura 3.12 este ilustrat modul în care un dispozitiv nou este adăugat în sistem. Această interacțiune este inițiată de către utilizator. Utilizatorul va trebui să selecteze ce dispozitiv dorește să adauge în caz că sunt prezente mai multe dispozitive pornite dar care nu se află în nici o rețea personală. Apoi aplicația va cere, ca utilizatorul să introducă cheia de acces către dispozitiv. Aplicația mobilă va transmite cererea de conectare împreună cu cheia de acces către dispozitiv. Acesta va evalua cheia și în dependență de corectitudinea acesteia va accepta sau refuza conexiunea. Utilizatorul va fi înștiințat de rezultatul acestei operații.

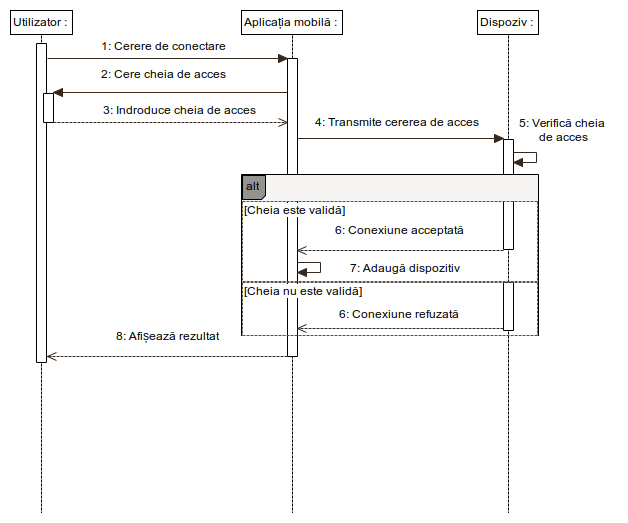


Figura 3.12 - Conectarea unui nou dispozitiv

### Fluxurile de mesaje şi legăturile dintre componentele sistemului

În figura 3.13 este prezentată colaborarea dintre componentele fizice ale dispozitivului pentru a colecta datele. În primul rând, microcontrolerul va inițializa senzorii digitali BME680 și LTR 390UV. Apoi când măsurarea datelor va fi finalizată va citi parametrii respectivi. De pe senzorul BME680 vor fi citite temperatura, umiditatea, presiunea și compușii volatili organici. De pe senzorul LTR 390UV va fi citit nivelul de iluminare și radiații ultra violete. În ultimul rând, microfonul ZTS6117 care este un senzor analog, va modifica tensiunea pe un pin, iar microcontrolerul o va citi. Apoi în urma unor calcule va fi identificat nivelul de gălăgie. Pentru a păstra datele, în cazul când nu există o conexiune Bluetooth, microcontrolerul va înscrie aceste date în regiunea de ram a RTC-ului, această regiune de RAM nu pierde alimentarea nici când microcontrolerul este în regim de somn, deci datele nu vor fi pierdute. Când va fi prezentă o conexiune, microcontrolerul va putea citi aceste date și să le transmite către aplicația mobilă.

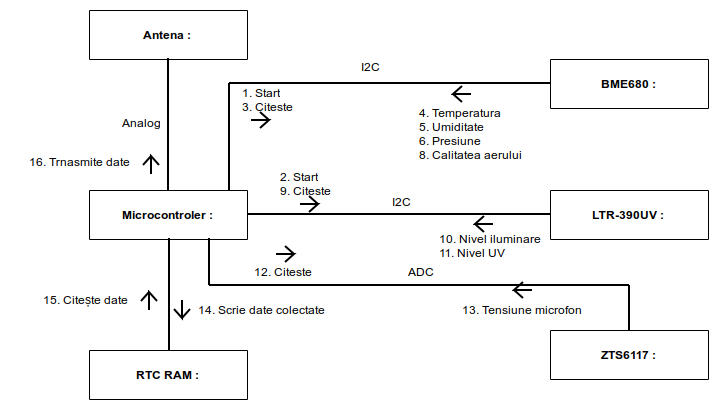


Figura 3.13 - Colectarea datelor

Pentru a adăuga un dispozitiv nou, utilizatorul trebuie să selecteze ce dispozitiv dorește să adauge în caz că sunt prezente mai multe dispozitive. Apoi aplicația va cere, ca utilizatorul să introducă cheia de acces către dispozitiv. Aplicația mobilă va transmite cererea de conectare împreună cu cheia de acces către dispozitiv. Acesta va evalua cheia și în dependență de corectitudinea acesteia va accepta sau refuza conexiunea. Utilizatorul va fi înștiințat de rezultatul acestei operații. Acest proces poate fi vizualizat sub formă de colaborare în figura 3.14.

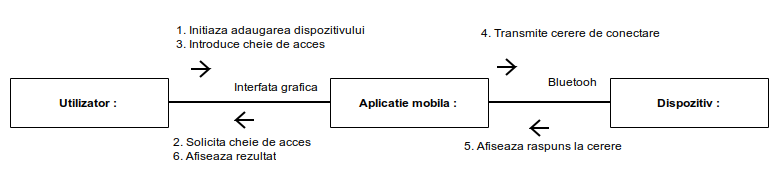


Figura 3.14 - Conectarea unui nou dispozitiv

## Descrierea structurală a sistemului

Pentru a modela structura unui sistem utilizând limbajul de modelare UML, pot fi utilizate următoarele diagrame:

* diagramă de clasă;
* diagramă de obiecte;
* diagramă de componente;
* diagramă de implementare.
* diagramă de structură compozită;
* diagramă de pachete;
* diagramă de profiluri.

Diagrama de clasă este ce-a mai utilizată diagramă de structură în dezvoltarea software. Deși în principal este utilizată pentru a reprezenta clase, poate fi utilizată și pentru a reprezenta structuri și enumerații pentru limbajele care nu au clase cum ar fi C. Cu acest tip de diagramă poate fi prezentat modelul logic și fizic al sistemului. Reprezentarea grafică a clasei este un dreptunghi cu trei secțiuni. În prima se regăsește numele și stereotipul clasei. În următoarea secțiune sunt prezentate atributele. În ultima secțiune sunt prezentate metodele clasei.

Diagrama de obiecte este utilizată pentru a instalația un exemplu al diagramei de clasă. Aceasta are ca scop să valideze corectitudinea diagramei de clasă.

Diagrama de componente prezintă grupările logice ale elementelor. Aici pot fi prezentate modulele software ale sistemului. Dar și componentele obiectelor fizice. Fiecare component este reprezentat cu ajutorul unei casete rectangulare. Conectorii sunt interfețe oferite sau cerute.

Diagrama de structură compozită este utilizată rareori în afara domeniului de dezvoltare software. Deoarece este mai detaliată decât diagrama de clase, descriind mai multe amănunte despre clase și interacțiunile dintre acestea. Pentru majoritatea cazurilor această diagramă este una în plus.

Diagrama de implementare prezintă componentele hardware și software ale sistemului. Componentele hardware sunt prezentate ca noduri iar componentele software sunt prezentate ca artefacte. Această diagramă prezintă modul de instalare al sistemului și cum componentele sistemului sunt legate.

Diagrama de pachete este utilizată pentru a prezenta dependențele dintre pachetele care compun un model. Scopul principal este de a prezenta relația dintre diferitele componente mari care formează un sistem complex.

Diagrama de profiluri este la fel de complexă ca un limbaj. Pentru această diagramă sunt create noi proprietăți și semnificații pentru diagramele UML existente. Aici se definesc noi stereotipuri. Profilurile sunt utile pentru a particulariza un model UML pentru o platformă specifică, cum ar fi Java sau .Net Framework.

### Descrierea structurii statice a sistemului

Pentru implementarea senzorilor la nivel de driver, indiferent de limbajul care va fi utilizat, poate fi utilizat șablonul de proiectare hardware adaptor [23]. În figura 3.15 este prezentat faptul că toți senzorii vor moșteni de la clasa senzor. Aceștia vor trebui să ofere o singură metodă pentru citirea parametrilor, însă aceasta va primi tipul parametrului care va fi citit. Astfel se va respecta și principiul substituirii Liskov [24].

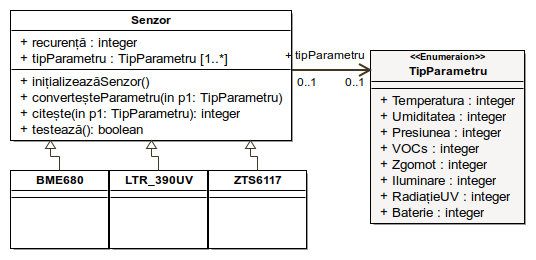


Figura 3.15 - Implementarea senzorilor

În cod va fi prezentă o clasă pentru generarea stărilor dispozitivului. În metodele acesteia se va scrie codul pentru inițializare, auto testare, operațiune și somn. Acestea vor fi implementate cu ajutorul mașinii de stări prezentată în figura 3.9. Cum poate fi observat în figura 3.16 această clasă va interacționa și cu alte clase cum ar fi clasa responsabilă de colectarea datelor, care va avea un șir de senzori. Și clasa de auto testare care va avea metode pentru a testa toate părțile componente ale dispozitivului.

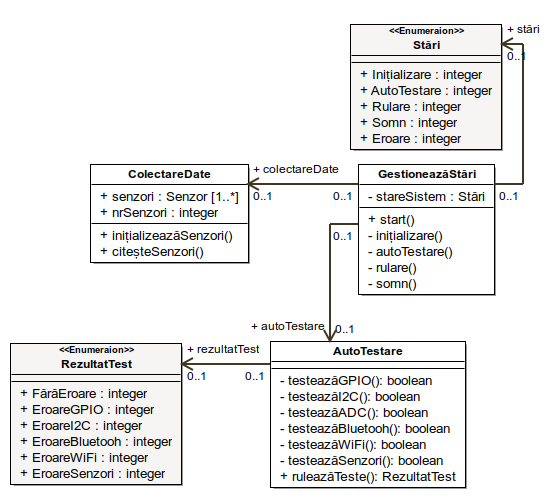


Figura 3.16 - Gestionarea stărilor dispozitivului

### Relațiile de dependență între componentele sistemului

Componentele software ale aplicației mobile sunt prezentate în figura 3.17. Acestea sunt logica de aplicație, baza de date, setări, prezentarea și comunicarea. Nivelul de prezentare va oferi o interfață pentru afișarea datelor. Nivelul de comunicare va permite stabilirea conexiunii cu dispozitivul, transmiterea de comenzi și preluarea datelor. Baza de date va permite stocarea datelor colectate pe termen lung. Componenta de setări va permite de a salva setările utilizatorului ca acestea să nu se piardă.

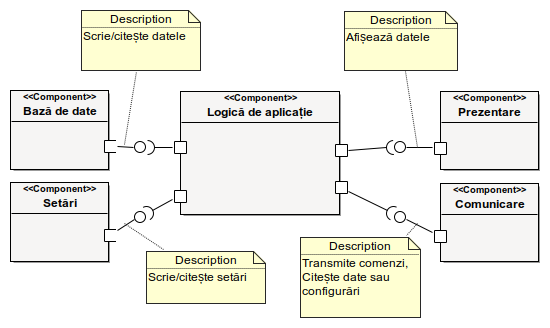


Figura 3.17 - Componentele aplicației mobile

În figura 3.18 sunt prezentate componentele software ale ale dispozitivului. Acestea includ urătoarele componente: aplicație, persistență, comunicare, auto testare, driver LTR 390UV, driver BME680, driver ZTS6117. Componenta de persistență va avea rolul de a stoca setările, în principal cele ce țin de recurența eșantionării parametrilor, conexiuni Wi-Fi și MQTT, dar și stocarea datelor pentru recuperarea lor ulterioară. Componenta de comunicare va permite conectarea la rețele Wi-Fi și Bluetooth, dar și transmiterea datelor prin aceste canale. Pentru transmiterea datelor pin WiFi se va folosi protocolul MQTT iar pentru transmiterea datelor prin Bluetooth se va implementa un protocol propriu. Componenta de auto testare va testa funcționalitatea dispozitivului și integritatea fiecărui senzor. Driverele pentru fiecare senzor va permite citirea datelor oferite de aceștia dar și identificarea dacă senzorul este în stare funcțională și nu are defecte.

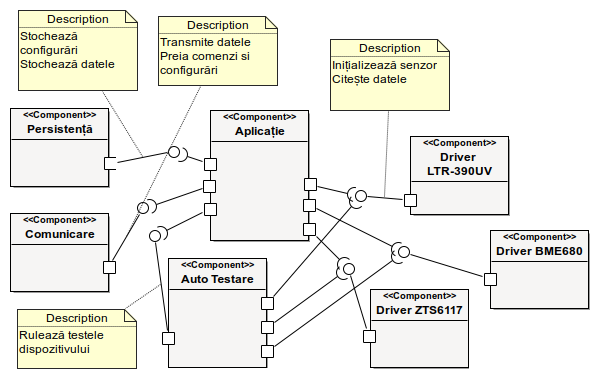


Figura 3.18 - Componentele software-ului dispozitivului

### Modelarea echipamentelor mediului de implementare

În figura 3.19 este prezentată implementarea sistemului în ansamblu. O instanță a sistemului este constituită din două tipuri de elemente: dispozitiv și aplicație mobilă. În cadrul acestei instanță, pot fi conectate mai multe dispozitive la o singură aplicație mobilă. Nucleul dispozitivului este microcontrolerul ESP32-C3, care va rula firmware-ul scris pentru acesta. La microcontroler vor fi conectați senzorii respectivi, BME680, ZTS6117 și LTR 390UV de pe care se vor citi parametrii de interes. Pe telefonul mobil, spre deosebire de dispozitiv este instalat și un sistem de operare, care va constitui mediul de execuție al aplicației. Acest sistem de operare va permite accesul aplicației la rețeaua Bluetooth, dar și la alte funcții, cum ar fi conexiunea la internet sau afișarea unei interfețe grafice. În cadrul acestui sistem pot fi conectate mai multe dispozitive. Totodată acest sistem ar putea să transmită sau să preia date de la un broker MQTT.

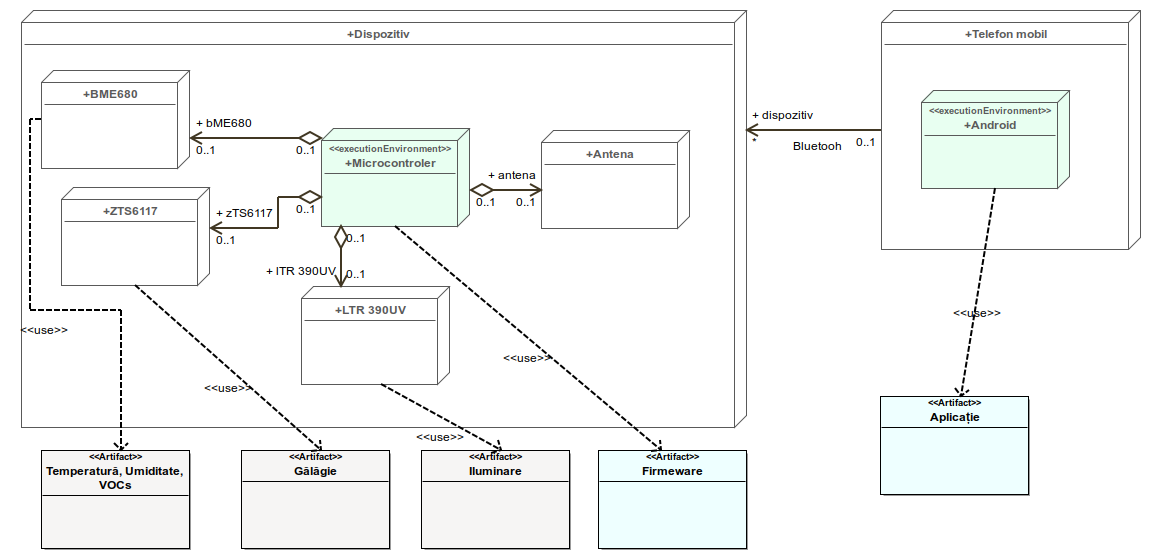


Figura 3.19 - Implementarea sistemului

# REALIZAREA SISTEMULUI

Se face o introducere în ce s-a folosit pentru realizarea sarcinilor. Se descrie pe scurt tehnologiile folosite, limbaje de programare utilizate, instrumente, algoritmi, șabloane...etc , totul ce s-a folosit la realizarea proiectului.

Pentru realizarea sistemului au fost nevoie de mai multe instrumente software. S-a optat ca toate aceste instrumente să fie libere și open spurce pentru a nu fi nevoie de a le pirata. În continuare este prezentată lista de software utilizată pentru a realiza sistemul:

* Gnu/Linux [25] – sistem de operare;
* terminal – pentru a accesa editorul de text, compilatoarele și alte programe;
* Neovim [26] – editor de text hiper-extensibil bazat pe Vim [27];
* flutter-tools.nvim [28] – extensie Neovim pentru crearea aplicațiilor flutter;
* Android Studio [29] – a fost utilizat emulatorul de android oferit de acesta;
* Git [30] – instrument pentru versionarea codului sursă;
* Github [31] – platformă online de găzduire a codului sursă.

Neovim este al doilea cel mai popular editor de text care rulează în terminal din anul 2023 [32]. Printre avantajele acestui editor este modul eficient de a edita textul fiind prezente mai multe moduri de editare, cum ar fi normal, insert, selectare vizuală și altele. Totodată acest editor are o comunitate activă care creează și mențin o mulțime de extensii utile. Acest editor poate fi adaptat pentru o mulțime de cazuri de utilizare, iar configurarea acestuia este realizată prin editarea unui fișier, care poate fi copiat și utilizat și pe alte calculatoare. Pentru a facilita dezvoltarea unei aplicații Flutter, a fost instalată extensia flutter-tools.nvim, care oferă colorarea sintaxei și alte funcționalități. Meniul acestui plugin poate fi vizualizat în figura 4.1.



Figura 4.1 - Meniul pluginului flutter tools

Fiind că aplicația mobilă este destinată de a rula pe telefoane mobile care de cele mai multe ori au procesoare ARM însă majoritatea calculatoarelor sunt pe platforma x86, a fost nevoie de a instala un emulator android pentru a permite testarea mai rapidă a schimbărilor de cod. Acest emulator a fost obținut instalând aplicația Android Studio.

Git este cel mai popular instrument pentru versionarea codului sursă [33]. Totodată el poate versiona și alte lucruri, nu doar codul sursă, spre exemplu documente sau imagini care vor fi modificate. Totuși este mult mai ușor de a lucra cu fișiere textuale decât cu fișiere binare în versionarea acestora, fiind posibilă compararea a două versiuni.

Github este o platformă online care permite găzduirea codului sursă, atât în repozitorii private cât și în repozitorii publice. Totodată oferă posibilitatea de a crea pipeline-uri pentru integrarea continuă și plasarea continuă, însă repozitoriile private au niște limitări în acest domeniu.

Pentru realizarea sistemului au fost utilizate mai multe limbaje de programare, compilatoare și librării. Limbajele de programare sunt C pentru utilizarea driverelor și secvențelor de cod existente pe internet. Zig [34] limbaj de programare și compilator care permite compilarea atât a codului Zig cât și C în același proiect. Microzig [35] strat de abstractizare unificat pentru mai multe microcontrolere, inclusiv ESP32C3. Limbajul Dart împreună cu framework-ul Flutter pentru crearea aplicației mobile.

Pentru dezvoltarea proiectului a fost utilizat un laptop Lenovo Thinkpad E595 cu procesor AMD Ryzen 7 3700U cu 24 GB de RAM. Dintre perifericele utilizate în-afară de monitorul, tastatura și touchpad-ul laptopului au mai fost folosite încă 2 monitoare externe. O tastatură ergonomică, despicată și ortoliniară Dactyl Manuform [36] și un trackball asemănător cu Logitech MX ERGO [37].

## Descrierea la nivel de cod pe module

Aici descrieți cum ați implementat funcționalul aplicației la nivel de cod , dacă ați folosit un algoritm îl descrieți.

### Aplicația mobilă

Prima pagină implementată a aplicației mobile a fost cea de pornire. Pe această pagină este prezentat titlul aplicației rotit cu 90 de grade. Această pagină va fi utilizată pentru a aștepta încărcarea datelor în aplicație. Implementarea vizuală a acestei pagini poate fi vizualizată în figura 4.2.



Figura 4.2 - Pagina de pornire

În figura 4.3 este prezentată o porțiune a paginii principale. Aceasta are o bară pe care în partea stânga se află pictograma setărilor iar în partea dreaptă se află pictograma alarmelor. Acest lucru va face atât setările cât și alarmele accesibile. Dedesubtul acestei bare se află o listă de dispozitive. Acestea vor fi dispozitivele înregistrate în sistem. Dispozitivele active for fi prezentate cu un disc de culoarea verde iar cele inactive cu un disc de culoarea roșie.

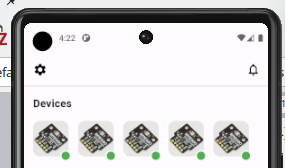


Figura 4.3 - Lista de dispozitive

Modul de a configura o alarmă este prezentat în figura 4.4. Utilizatorul poate introduce un titlu pentru această alarmă, care va fi afișat câmd este declanșată alarma. În cadrul alarmei curente, utilizatorul va putea indica ce parametru dorește să fie monitorizat. Pentru fiecare parametru poate fi specificat un interval, și modul în care se dorește evaluarea parametrului curent. Dacă aflarea acestuia în interiorul sau exteriorul parametrului va declanșa alarma.

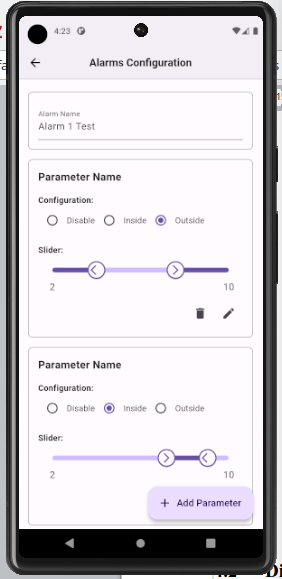


Figura 4.4 - Configurarea unei alarme

Pe pagina cu setări se regăsește un link către pagina de Github a proiectului, aceasta poate fi vizualizată în figura 4.5. Următorul element este un comutator care activează verificarea automată a noilor versiuni software pentru aplicație, următorul comutator verifică dacă există noi actualizări de firmware pentru dispozitiv. Următorul element din meniu va deschide o pagină pentru configurarea recurenței implicite a parametrilor însă utilizatorul va avea posibilitatea să decidă dacă dorește aceasta să fie utilizată pe fiecare dispozitiv în parte. Următorul element va permite de a adăuga conexiuni noi Wi-Fi în cazul când se va dori ca dispozitivul să aibă acces la internet. Apoi în ultimul element vor putea fi setate acreditarea pentru conectarea la un serviciu MQTT.

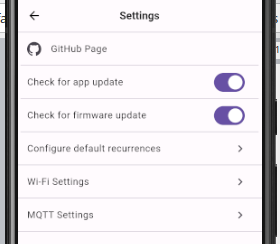


Figura 4.5 - Setările aplicației

Pagina de detalii, va mai trebui să fie actualizată pentru a arăta mai bine. Cel puțin la moment aceasta va putea afișa datele care sunt colectate. Implementarea aceste pagini poate fi vizualizată în figura 4.6.

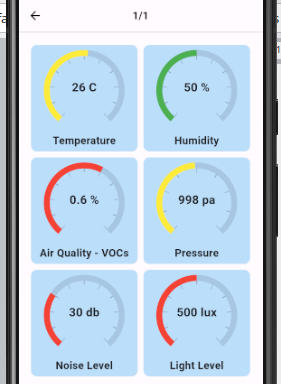


Figura 4.6 - Pagina de detalii

Un detaliu important despre realizarea aplicației este că a fost adăugată localizarea. La momentul aplicația suportă doar limba engleză, dar după necesitate fișierul app\_en.arb poate fi tradus în alte limbi cum ar fi româna. Acest lucru este reprezentat în figura 4.7.

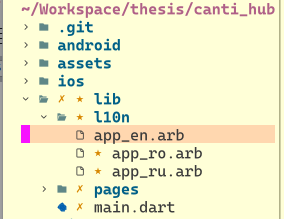


Figura 4.7 - Localizarea

În figura 4.8 este prezentat istoricul de mesaje din sistemul Git. La moment au fost făcute 75 de comiteri, care respectă convenția de comiteri **conventional commit.** Aplicația este aproape finalizată.

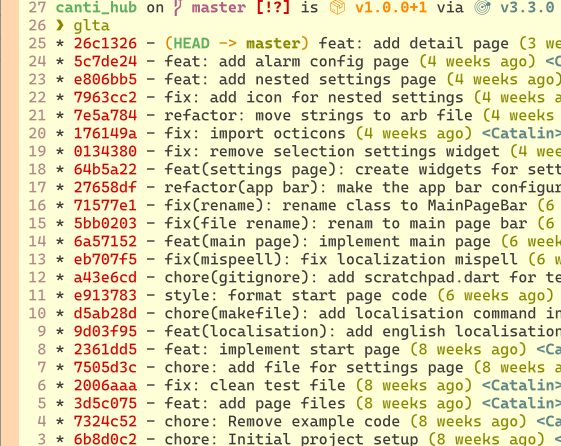


Figura 4.8 - Istoricul de mesaje

## Testarea sistemului

În acest subcapitol descrieți cum ați testat ce ați realizat. Descrieți metoda de testare, ce instrumente ați folosit, care au fost rezultate testării și cum ați soluționat problemele apărute.

# DOCUMENTAREA PRODUSULUI REALIZAT

Descrieți cerințele tehnice pentu aplicație, cum se gasește aplicația, cum se instalează.

În acest capitol se descrie utilizarea aplicației la nivel de administrator și la nivel de utilizator foarte detaliat, ca un manual de utilizare a aplicației.

Descrierea fiecărui butonaș, meniu, toate funcțiile.

Cum se obtine sistemul

aplicația se descarca de pe github

dispozitivul se poate comanda de pe jlc pcb

de instalat firmware

Aplicația are doar un mod de lucru (utilizator simplu, care este și administratorul)

descrierea fiecarei pagini fiecare buton cu linii și cifre de scris pe imagine ce reprezintă fiecare buton

Descrierea butoanelor dispozitivului, pe ce se apasa, cum se pune în priza,

# ESTIMAREA COSTURILOR PROIECTULUI

Acest capitol conține estimarea costurilor pentru realizarea proiectului. Pentru a estima costurile cu o acuratețe mare s-a optat pentru metoda de descompunere a lucrărilor – WBS. Această metodă presupune descompunerea sarcinilor pentru realizarea proiectului, optimal fiecare element trebuie să fie descompus în cel mult 5 elemente. Se recomandă ca elementele de bază să aibă o durată de realizare între o zi de muncă și două săptămâni.

Diagrama WBS cu sarcinile necesare pentru finisarea proiectului poate fi văzută în figura 6.1 iar estimarea costurilor și timpul de executare al sarcinilor poate fi observat în tabelul A.1. Etapele principale în realizarea sistemului sunt:

* analiza domeniului;
* modelarea și proiectarea;
* realizarea sistemului;
* documentarea sistemului.

În total pentru realizarea sistemului s-a estimat costuri de 9816 $, aceste costuri sunt estimate doar pentru munca depusă de presupușii angajați. În total a fost estimată realizarea proiectului în 91 de zile de lucru. Iar pentru fiecare etapă a fost considerat un salariu diferit în funcție de specialistul necesar, însă salariile nu diferă considerabil specialiști care nu sunt foarte calificați și nu au multă experiență.

Costurile pentru analiza domeniului constituie 17% din suma totală; inițial s-a estimat că această etapă va fi realizată în 16 zile. Modelarea proiectului constituie 23% din costurile totale, iar durata de realizare a acesteia a fost estimată la 21 de zile. Fără îndoială, realizarea sistemului este cea mai costisitoare etapă a proiectului; aceasta necesită 59% din costurile totale, efectuându-se într-un interval de 52 de zile. Dintre aceste costuri, surprinzător, pentru aplicația mobilă se necesită 59% din costurile pe realizare și se efectuează într-o durată de 31 de zile. Acest fapt se datorează dificultății cu care programatorul a implementat interfața grafică, mai mare decât cea pentru dispozitiv. Pentru realizarea dispozitivului sunt necesare celelalte 41% din costuri și o durată de 18 zile. Fiindcă sistemul este relativ simplu, s-a estimat că documentarea acestuia va putea fi efectuată în 2 zile, iar costurile pentru această sarcină vor constitui 1% din costurile totale.

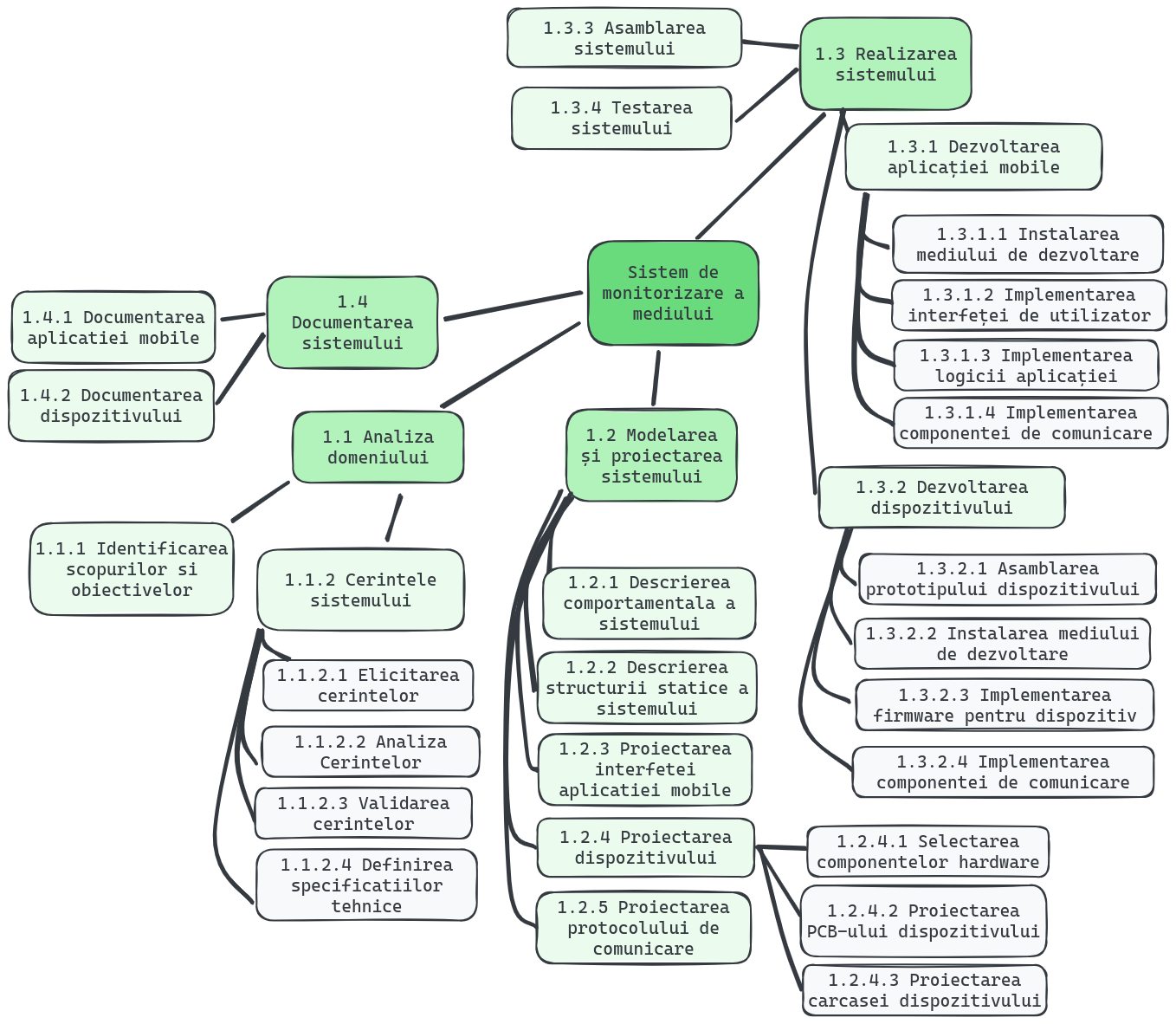


Figura 6.1 - Diagrama WBS.

În concluzi, costurile pentru realizarea unui proiect similar pot fi reduse drastic atunci când specialiștilor specialiștii sunt plătiți cu un salariu mic. Totodată durata de realizare a proiectului poate fi redusă dacă pentru realizare pot fi alocate să lucreze mai multe persoane în paralel.

CONCLUZII

Aici scrieți concluzii pentru tot raportul ( pentru fiecare capitol din raport)

În cadrul acestei practici a fost efectuate o mare parte din eforturile necesare pentru finisarea proiectului. În principal eforturile ce țin de realizarea raportului. În primul rând a fost analizat domeniul de monitorizare al mediului din care face acest proiect. Și au fost comparate trei sisteme existente asemănătoare cu sistemul dezvoltat, Atmotube PRO, Garmin Tempe și DT-8820. A fost realizate specificațiile sistemului. Și a fost proiectat sistemul. De asemenea s-a început lucrul asupra realizării sistemului.

În urma lucrărilor efectuate, interfața aplicației mobile este aproape finisată, la ea mai trebuie doar de îmbunătățit pagina de detalii, de adăugat tema luminoasă și întunecată adaptivă și de implementat legătura între pagini. La moment obiectivele ce țin de aplicația mobilă sunt parțial îndeplinite find că aplicația nu este într-o stare funcțională.

A fost observat faptul că este mult mai ușor să lucrezi la un proiect care ești plătit să îl realizezi decât la unul pe care îl faci din inițiativă proprie. Fără de a fi motivat de aspectul material. Altă posibilitate este că munca în echipă este mai benefică decât munca solo fiind că membri echipei te motivează și îți aduc aminte că trebuie să îți faci sarcinile.

Această practică a avut o importanță dubioasă, în cadrul practicii a fost o pauză de două săptămâni care a făcut dificlă reînceperea lucrărilor în a doua perioadă. De asemenea faptul că se cere de a realiza un videoclip ca rezultat al acestei practici iarăși, împiedică dedicarea eforturilor pentru finisarea proiectului.

**BIBLIOGRAFIE**

[1] D. E. Greenfield, „Environmental Monitoring: A Complete Guide”, Sigma Earth. Data accesării: 26 februarie 2024. [Online]. Disponibil la: https://sigmaearth.com/environmental-monitoring-a-complete-guide/

[2] „What are some of the current trends and challenges in environmental monitoring?” Data accesării: 23 noiembrie 2023. [Online]. Disponibil la: https://www.linkedin.com/advice/1/what-some-current-trends-challenges-environmental

[3] „All about Internet of Things (#IoT)”. Data accesării: 26 februarie 2024. [Online]. Disponibil la: https://www.linkedin.com/pulse/all-internet-things-iot-ahmed-karam

[4] „What Is Information Technology? A Beginner’s Guide to the World of IT”, Rasmussen University. Data accesării: 26 februarie 2024. [Online]. Disponibil la: https://www.rasmussen.edu/degrees/technology/blog/what-is-information-technology/

[5] R. Christopher, *Civilized to Death: The Price of Progress*. Data accesării: 7 aprilie 2024. [Online]. Disponibil la: https://www.amazon.com/Civilized-to-Death-audiobook/dp/B07QGH3TDF/ref=sr\_1\_1?crid=CONMXH5XS74S&dib=eyJ2IjoiMSJ9.6IB6Bz9DlPHDd0kODBnXVwx8vSV934bt9iF8AnLkQHI54wdKsRvEREv6erXxL8Ca3R4uQ1Jwj9P4AqTrayoF1uirKC4n\_FNHXiSO6WZLHGCkc093lf0jP4pR3XuO88kWVAnmF4pEjfUAiYAyDFjqP8sUxcphX0CBmb8EVobV7PrdeC2UyqZWO0nocGq2VXf4xVamtjKn-s2uxS60ioAhYWCcNm8gjssjaw3dhBPaRh8.XeQ-nYkYzOocYUiSN1\_H\_26Col8El-luqRg5\_QN8pWk&dib\_tag=se&keywords=civilized+to+death&qid=1712518477&sprefix=civilized+to+%2Caps%2C321&sr=8-1

[6] R. Fuller *et al.*, „Pollution and health: a progress update”, *Lancet Planet. Health*, vol. 6, nr. 6, pp. e535–e547, iun. 2022, doi: 10.1016/S2542-5196(22)00090-0.

[7] D. Khovalyg *et al.*, „Critical review of standards for indoor thermal environment and air quality”, *Energy Build.*, vol. 213, p. 109819, apr. 2020, doi: 10.1016/j.enbuild.2020.109819.

[8] „Atmotube PRO - Wearable and portable air quality monitor”. Data accesării: 23 noiembrie 2023. [Online]. Disponibil la: https://atmotube.com/atmotube-pro

[9] „tempeTM - Garmin Moldova”. Data accesării: 23 noiembrie 2023. [Online]. Disponibil la: https://garmin.md/produs/010-11092-30/

[10] „DT-8820 - 4 in 1 Multifunction Environment Meter”. Data accesării: 21 februarie 2024. [Online]. Disponibil la: https://www.cem-instruments.com/en/product-id-929

[11] „An Overview of the IoT Tech Stack | IoT Glossary”. Data accesării: 23 noiembrie 2023. [Online]. Disponibil la: https://www.emnify.com/iot-glossary/iot-stack

[12] Russell, „FPGA vs. Microcontroller vs. ASIC”, Nandland. Data accesării: 28 februarie 2024. [Online]. Disponibil la: https://nandland.com/lesson-2-fpga-vs-micro-vs-asic/

[13] M. Lab, „Bare-metal and RTOS Based Embedded Systems”, Microcontrollers Lab. Data accesării: 29 februarie 2024. [Online]. Disponibil la: https://microcontrollerslab.com/difference-bare-metal-and-rtos-based-embedded-systems/

[14] „Why I rewrote my Rust keyboard firmware in Zig: consistency, mastery, and fun”. Data accesării: 29 februarie 2024. [Online]. Disponibil la: https://kevinlynagh.com/rust-zig/

[15] „A 2022 Guide to IoT Protocols and Standards”, Particle. Data accesării: 29 februarie 2024. [Online]. Disponibil la: https://www.particle.io/iot-guides-and-resources/iot-protocols-and-standards/

[16] „HG353/2010 cu privire la aprobarea cerinţelor minime de securitate şi sănătate la locul de muncă”. Data accesării: 20 februarie 2024. [Online]. Disponibil la: https://www.legis.md/cautare/getResults?doc\_id=22129&lang=ro

[17] „Regulamentul sanitar privind normativele de emitere a zgomotului și a vibrației”. Data accesării: 20 februarie 2024. [Online]. Disponibil la: https://gov.md/sites/default/files/document/attachments/intr07\_1\_18.pdf

[18] M. Gulin, „Answer to «Body Energy Consumption: calculating watts and kwh given calories?»”, Physics Stack Exchange. Data accesării: 27 februarie 2024. [Online]. Disponibil la: https://physics.stackexchange.com/a/702472

[19] „AADL resource pages”. Data accesării: 1 martie 2024. [Online]. Disponibil la: http://www.openaadl.org/

[20] „Unified Modeling Language, v2.5.1”, *Unified Model. Lang.*.

[21] „Modelio Open Source - UML and BPMN free modeling tool”. Data accesării: 1 martie 2024. [Online]. Disponibil la: https://www.modelio.org/index.htm

[22] „Ghidul simplu pentru diagrame UML și modelarea bazelor de date”. Data accesării: 1 martie 2024. [Online]. Disponibil la: https://www.microsoft.com/ro-ro/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling

[23] B. P. Douglass, *Design patterns for embedded systems in C: an embedded software engineering toolkit*, 1st ed. Oxford ; Burlington, MA: Newnes/Elsevier, 2011.

[24] „SOLID: The First 5 Principles of Object Oriented Design | DigitalOcean”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design

[25] „Linux and GNU - GNU Project - Free Software Foundation”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://www.gnu.org/gnu/linux-and-gnu.en.html

[26] „Home - Neovim”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://neovim.io/

[27] „welcome home : vim online”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://www.vim.org/

[28] Akin, „akinsho/flutter-tools.nvim”. 2 martie 2024. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://github.com/akinsho/flutter-tools.nvim

[29] „Download Android Studio & App Tools”, Android Developers. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://developer.android.com/studio

[30] „Git”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://git-scm.com/

[31] „Build software better, together”, GitHub. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://github.com

[32] „Stack Overflow Developer Survey 2023”, Stack Overflow. Data accesării: 23 noiembrie 2023. [Online]. Disponibil la: https://survey.stackoverflow.co/2023/?utm\_source=social-share&utm\_medium=social&utm\_campaign=dev-survey-2023

[33] „Stack Overflow Developer Survey 2022”, Stack Overflow. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://survey.stackoverflow.co/2022/?utm\_source=social-share&utm\_medium=social&utm\_campaign=dev-survey-2022

[34] „Home ⚡ Zig Programming Language”. Data accesării: 21 februarie 2024. [Online]. Disponibil la: https://ziglang.org/

[35] „ZigEmbeddedGroup/microzig”. Zig Embedded Group, 3 martie 2024. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://github.com/ZigEmbeddedGroup/microzig

[36] J. Bertrand, „abstracthat/dactyl-manuform”. 2 martie 2024. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://github.com/abstracthat/dactyl-manuform

[37] „Logitech MX ERGO Advanced Wireless Trackball with Tilt Plate”. Data accesării: 3 martie 2024. [Online]. Disponibil la: https://www.logitech.com/en-eu/products/mice/mx-ergo-wireless-trackball-mouse.html

ANEXA A

Estimarea Costurilor

Tabelul A.1 – Estimarea costurilor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WBS** | **Denumire** | **Început** | **Sfârșit** | **Lucru** | **Cost** |
| **1** | **Sistem de monitorizare a mediului** | **Sep 7** | **Mai 13** | **91d** | **9816** |
| **1.1** | **Analiza domeniului** | **Sep 7** | **Sep 26** | **16d** | **1664** |
| 1.1.1 | Identificarea scopurilor si obiectivelor | Sep 7 | Sep 7 | 1d | 104 |
| **1.1.2** | **Cerințele sistemului** | **Sep 8** | **Sep 26** | **15d** | **1560** |
| 1.1.2.1 | Elicitarea cerințelor | Sep 8 | Sep 14 | 5d | 520 |
| 1.1.2.2 | Analiza cerințelor | Sep 15 | Sep 20 | 4d | 416 |
| 1.1.2.3 | Validarea cerințelor | Sep 19 | Sep 22 | 4d | 416 |
| 1.1.2.4 | Definirea specificațiilor tehnice | Sep 25 | Sep 26 | 2d | 208 |
| **1.2** | **Modelarea și proiectarea sistemului** | **Oct 3** | **Mai 1** | **21d** | **2184** |
| 1.2.1 | Descrierea comportamentala a sistemului | Nov 1 | Nov 10 | 8d | 832 |
| 1.2.2 | Descrierea structurii statice a sistemului | Nov 13 | Nov 17 | 5d | 520 |
| 1.2.3 | Proiectarea interfeței aplicației mobile | Oct 3 | Oct 5 | 3d | 312 |
| **1.2.4** | **Proiectarea dispozitivului** | **Oct 5** | **Apr 24** | **4d** | **416** |
| 1.2.4.1 | Selectarea componentelor hardware | Oct 5 | Oct 5 | 1d | 104 |
| 1.2.4.2 | Proiectarea PCB-ului dispozitivului | Apr 22 | Apr 23 | 2d | 208 |
| 1.2.4.3 | Proiectarea carcasei dispozitivului | Apr 24 | Apr 24 | 1d | 104 |
| 1.2.5 | Proiectarea protocolului de comunicare | Mai 1 | Mai 1 | 1d | 104 |
| **1.3** | **Realizarea sistemului** | **Jan 15** | **Mai 10** | **52d** | **5808** |
| **1.3.1** | **Dezvoltarea aplicației mobile** | **Jan 15** | **Mai 7** | **31d** | **3472** |
| 1.3.1.1 | Instalarea mediului de dezvoltare | Jan 15 | Jan 15 | 1d | 112 |
| 1.3.1.2 | Implementarea interfeței de utilizator | Jan 29 | Feb 26 | 21d | 2,352 |
| 1.3.1.3 | Implementarea logicii aplicației | Mar 7 | Mar 15 | 7d | 784 |
| 1.3.1.4 | Implementarea componentei de comunicare | Mai 6 | Mai 7 | 2d | 224 |
| **1.3.2** | **Dezvoltarea dispozitivului** | **Apr 11** | **Mai 3** | **18d** | **2016** |
| 1.3.2.1 | Asamblarea prototipului dispozitivului | Apr 11 | Apr 11 | 1d | 112 |
| 1.3.2.2 | Instalarea mediului de dezvoltare | Apr 12 | Apr 12 | 1d | 112 |
| 1.3.2.3 | Implementarea firmware pentru dispozitiv | Apr 15 | Mai 2 | 14d | 1,568 |
| 1.3.2.4 | Implementarea componentei de comunicare | Mai 2 | Mai 3 | 2d | 224 |
| 1.3.3 | Asamblarea sistemului | May 8 | May 8 | 1d | 112 |

Continuare tabelul A.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WBS** | **Denumire** | **Început** | **Sfârșit** | **Lucru** | **Cost** |
| 1.3.4 | Testarea sistemului | May 9 | May 10 | 2d | 208 |
| **1.4** | **Documentarea sistemului** | **May 10** | **May 13** | **2d** | **160** |
| 1.4.1 | Documentarea aplicației mobile | May 10 | May 10 | 1d | 80 |
| 1.4.2 | Documentarea dispozitivului | May 13 | May 13 | 1d | 80 |