**Opportunistic Sensing Systems**

LUCRARE DE LICENŢĂ

|  |  |  |
| --- | --- | --- |
|  | Absolvent: | **Aurel-Ioan Vidrean** |
|  |  |  |
|  | Coordonator științific: | **titlul științific Bogdan Iancu** |

**2023**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
| DECAN, |  | | DIRECTOR DEPARTAMENT, | |
| **Prof. dr. ing. Liviu MICLEA** |  | **Prof. dr. ing. Rodica POTOLEA** | |

Absolvent: **Aurel-Ioan Vidrean**

**Opportunistic Sensing Systems**

1. **Enunțul temei:** *Scurtă descriere a temei lucrării de licență și datele inițiale*
2. **Conținutul lucrării:** *(enumerarea pârților componente) Exemplu: Pagina de prezentare, aprecierile coordonatorului de lucrare, titlul capitolului 1, titlul capitolului 2,… titlul capitolului n, bibliografie, anexe.*
3. **Locul documentării**: *Exemplu*: Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți**:
5. **Data emiterii temei:** 1 noiembrie 2022
6. **Data predării:** 8 iulie 2023

|  |  |
| --- | --- |
| Absolvent: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Coordonator științific: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Declarație pe propria răspundere privind**

**autenticitatea lucrării de licență**

Subsemnatul(a)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**, legitimat(ă) cu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ seria \_\_\_\_\_\_\_ nr. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
CNP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, autorul lucrării \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ din cadrul Universității Tehnice din Cluj-Napoca, sesiunea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a anului universitar \_\_\_\_\_\_\_\_\_\_, declar pe propria răspundere că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

In cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

|  |  |  |
| --- | --- | --- |
| Data  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Nume, Prenume  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
|  |  | Semnătura |

**Instrucțiuni generale**

**De citit înainte** (paginile de aici până la cuprins se vor elimina din versiunea finală):

1. Cele trei pagini anterioare (foaie de capăt, foaie sumar, declarație) se vor lista pe foi separate (nu fată-verso), fiind incluse în lucrarea listată.
   1. Foaia de sumar (a doua) necesită semnătura absolventului, respectiv a coordonatorului.
   2. Pe declarație se trece data când se predă lucrarea la secretarii de comisie.
   3. Pe foaia de capăt, se va trece corect titulatura cadrului didactic îndrumător (consultați pagina de unde ați descărcat acest document pentru lista cadrelor didactice cu titulaturile lor).
2. **Cuprinsul** începe pe pagina nouă, impară (dacă se face listare față-verso). Pentru actualizarea cuprinsului, folosiți meniul *References: Table of Contents->Update table*. Cuprinsul este numerotat cu cifre romane, litere mici.
3. Toate capitolele încep pe o pagină nouă. Acest lucru este realizat folosind simbolul de formatare *Section Break* cu opțiunea *Next Page*, care a fost folosit la sfârșitul fiecărui capitol. Dacă ștergeți din greșeală simbolul, îl puteți pune înapoi folosind submeniul *Breaks* al meniului *Layout*.

Numerotarea paginilor începe de la 1 la primul capitol (**Introducere)**.

1. **Formatarea:**
   1. Mărimea paginii este A4, cu toate marginile de 25.4 mm
   2. Fontul folosit implicit în acest document este Times New Roman, 12pt, conform stilului *Normal*, cu spațiere la 1 rând (*Paragraph*->*Line spacing->* 1.0) și Alignment: *Justify***.**
   3. Capitolele încep pe o pagină nouă.
   4. Pentru prima linie din fiecare paragraf se folosește indentare (implicit in *Normal Style*), iar între paragrafe succesive nu se lasă distanță suplimentară.
   5. Folosiți stilurile predefinite în acest document (Headings, Figura, Tabel, Normal etc.)
   6. **Titlurile**:
      1. Pentru capitole: Heading 1, numerotat cu o cifra (Capitolul x. Numele capitolului), font Times New Roman 16pt, Bold, 30pt spațiu vertical înainte, 18pt spațiu după.
      2. Pentru secțiuni: Heading 2, font Times New Roman 14pt, Bold, spațiu vertical înainte de titlu 12pt, spațiu după 6pt.
      3. Pentru subsecțiuni: Heading 3, Times New Roman 12pt, Regular, spațiu vertical înainte de titlu 12pt, spacing după 6pt
2. Vizualizați (și în timpul editării, dacă e cazul) acest document după ce activați vizualizarea simbolurilor ascunse de formatare (click pe pictograma **** din meniul *Home->Paragraph*).
3. Respectați restul instrucțiunilor din fiecare capitol.
4. Înainte de salvarea ca fișier pdf, din meniul *Review* alegeți *No Markup*. Apoi alegeți din meniul *Options* opțiunile din figura următoare



Cuprins

[Capitolul 1. Introducere 1](#_Toc129864352)

[Capitolul 2. Obiectivele proiectului 2](#_Toc129864353)

[Capitolul 3. Studiu bibliografic 4](#_Toc129864354)

[Capitolul 4. Analiză și fundamentare Teoretică 6](#_Toc129864355)

[4.1. Exemplu de titlu de secțiune 6](#_Toc129864356)

[4.1.1. Exemplu de titlu de subsecțiune 6](#_Toc129864357)

[Capitolul 5. Proiectare de detaliu și implementare 8](#_Toc129864358)

[Capitolul 6. Testare și validare 10](#_Toc129864359)

[Capitolul 7. Manual de instalare si utilizare 12](#_Toc129864360)

[Capitolul 8. Concluzii 14](#_Toc129864361)

[Bibliografie 16](#_Toc129864362)

[Anexa 1....................................................................................................18](#_Toc129864363)

[Anexa 2................ 19](#_Toc129864364)

[Anexa 3......... 20](#_Toc129864365)

# Introducere – Contextul proiectului

**1.1 Dezvoltarea dispozitivelor mobile**

Tehnologia dispozitivele mobile a revolutionat modul in care ne desfasuram acitvitatile si felul in care traim, deoarece adoptarea dispozitivelor mobile a cunoscut o crestere semnificativa si rapida pe tot globul [[1]](#footnote-1). Telefoanele in sine au devenit insturmente ce faciliteaza cominucarea rapida si o implicare colectiva. Diferite grupuri de persoane le folosesc in diverse moduri creative. Totodata, numarul senzorilor incoporati in teelefoane si numarul aplicatilor construite in jurul acestora a crescut vertiginos. In ultimii ani, telefoanele mobile au inceput sa contina senzori precum GPS, accelerometru, giroscop, microfon, camera, Bluetooth, etc. Astfel, diverse aplicatii si servicii acopera anumite necesitati precum accesul la informatie, divertisment.

Omniprezenta dispozitivelor mobile si cresterea valorilor datelor generate de catre senzori si aplicații oferta un nou domeniu de cercetare in stiintele comportamentale si sociale. Cercetatorii au inceput sa examineze probleme in comportamentul utilizatorilor din perspectiva Big Data, folosind un volum mare de date pentru a caracteriza si intelege diverse comportamente precum trasaturi individuale, mobilitatea umana, comunicarea si modele de interactiuni [[2]](#footnote-2) [[3]](#footnote-3) [[4]](#footnote-4).

Oportunistic Sensing Systems (OSS) a devenit o arie de cercetare in crestere in stiinta calculatoarelor si inginerie. Sistemele utilizeaza dispozitive mobile si senzorii incorporati pentru a colecta, analiza si procesa datele in timp real, astfel oferind o noua metoda de a monitoriza mediul inconjurator, evenimentele ce se petrec in jur si pentru a oferi diverse informatii catre utilizatori.

Cresterea preocuparii pentru impactul poluarii fonice asupra sanatatii si bunastarii a dus la dezvoltarea a numeroase aplicatii ce monitorizeaza mediul inconjurator, incluzand sisteme fixe sau mobile de monitorizarea a masurarii zgomotului. Poluarea fonica este datorata mai multor factori precum transportul si activitatile industriale. Aceasta afecteaza atat mediul inconjurator (flora si fauna) cat si persoanele ce sunt expuse in mod repetat la acest fenomen, iar din pacate in orase acest lucru se resimte din ce in ce mai mult. Expunerea indelungata la poluarea fonica poate cauza pierderea auzului, hipertensiune si tulburari de somn. Totodata poate fi afectata si sanatatea mintala cauzand astfel stres, anxietate, chiar si depresie. Pe de alta parte acest fenomen poate afecta si calitatea vietii din anumite zone, deoarece fiind amplasate in locuri unde zgomotul este prezent constant vor exista pierderi din punct de vedere al preturilor locuintelor in zonele respective si activitatea economica poate fi afectata. Pentru a combate efectele negative ale poluarii fonice pot fi luate mai multe masuri precum bariere de zgomot, izolarea sunetului, reducerea zgomotului produs de transport si de activitatile industriale. Acest lucru se poate realiza cu ajutorul unor sisteme ce masoara concentratile de zgomot din diferite zone ale intinderilor urbane. Din pacate, aceste sisteme sunt foarte costisitoare, necesita o infrastructura foarte mare si totodata nu sunt usor de accesat pentru public. Aici intervine o solutie si anume integrarea unor astfel de sisteme la indemana oamenilor chiar pe dispozitivele lor mobile. Dispozitivele mobile, precum smartphone-uri si tablete, ofera o alternativa accesibila din punct de vedere al costurilor pentru monitorizarea mediului inconjurator.

**1.2 Detectia poluarii fonice cu ajutorul dipozitivelor mobile**

Acest proiect se focuseaza pe implementarea unei aplicatii Android care sa funcționeze pe post de Oportunistic Sensing System. Aplicatia o sa poata detecta nivelul de poluare fonica si prezenta persoanelor si a masinilor din jur in timp real folosind microfonul incorporat al dispozitivului. Proiectul urmareste sa atraga atentia asupra impactului pe care poluarea fonica o are asupra sanatatii si astfel oferind utilizatorilor date in timp real despre mediul inconjurator. Pentru functinoare aplicatia are nevoie de microfonul si GPS-ul incorporat al dispozitivului pentru a colecta date despre mediu si pentru a le procesa ulterior prin anumiti algoritmi de machine learning.

Acestea sunt echipate cu o varietate de senzori, incluzand microfon si GPS, ce pot fi folositi pentru a monitoriza mediul inconjurator. Pentru a face asta trebuie prelucrate datele ce survin de la aceste dispozitive, centralizate, prelucrate, iar apoi oferindu-se raporturi amanuntite in urma analizei facute in acest scop, raport public pentru orice utilizator al unei astfel de aplicatii.

Contextul proiectului este de a implementa o aplicatie de tipul Oportunistic Sensing Systems care sa ruleze pe Android. Fiind un sistem open-source ce faciliteaza construirea de diverse aplicatii si fiind la scara mondiala cel mai raspandit sistem de operare in randul dispozitivelor mobile am ales sa implementez o astfel de solutie pentru a combate problema descrisa mai sus.

Pentru realizarea obiectivului trebuie luate in considerare urmatoarele aspecte. Senzorii dispozitivelor mobile sunt calibrati intr-un anume fel si trebuie sa existe o calibrare a acestora pentru a interpreta cat mai corect datele colectate.

Pentru a realiza obiectivul proiectului trebuie sa fie implementati urmatorii pasi. In primul rand arhitectura aplicatiei si interfara pentru utilizator trebuie sa fie concepute si implementate.

# Obiectivele proiectului

In acest capitol se prezinta in detaliu motivatia care a dus la alegerea acestei teme, obiectivul principal si o serie de obiective secundare, cerintele functionale si non-functionale care trebuie indeplinite in urma implementarii solutiei propuse.

**2.1 Motivație**

Modul accelerat in care urbanizarea se dezvolta peste tot a determinat ca poluarea fonica sa devina o problema prevalenta si detrimentală. Pe masura ce orasele cresc si dezvolta, poluarea fonica isi face si ea simtita prezenta. De aceea o aplicatie care sa monitorizeze si sa inregistreze poluarea fonica este necesara pentru bunastarea comunitatilor.

**2.2 Obiective**

2.2.1 Obiectivul principal

Obiectivul principal este dezvoltarea unei aplicații care permite **detecția** polurii fonice la nivel local, **înregistrarea** valorilor detectate si **indicarea** valorilor din toate zonele aflate sub investigare. In acelasi timp, dorim ca prin intermediul acestei aplicatii sa putem detecta si concentrația autovehiculelor aflate in zona si a persoanelor de asemenea, astfel oferind o aplicatie mobila, portabila si cu un grad de performanta cat mai ridicat.

2.2.2 Obiective Secundare

Obiectivele secundare pe care as dori sa le ating pe parcursul dezvoltarii aplicatiei sunt: studierea solutiilor existente deja, definirea modului in care se va efectua analiza datelor, definirea consumului de energie necesar pentru prelucrarea datelor, analiza pe device-uri low-end si high-end.

**Studiere soluțiilor existente**

Identificare si intelegerea efectivitatii si limitarilor abordarilor curente duce la intelegerea conceptelor si tehnologiilor folosite. Analiza ofera mai multe perspective ce pot fi ulterior transformate in strategii de succes. Prin urmare aceasta etapa necesita o atentie sporita, influentand rezultatul final.

**Identificarea datelor necesare**

Pentru realizarea unei astfel de aplicații, este necesara o analiza a datelor provenite de la senzorii dispozitivului mobil pentru realizarea unui sistem utilizabil si fiabil. Odată facuta analiza va fi nevoie de un field de stocare de tipul [`noise\_detection`,`timestamp`], unde noise\_detection reprezinta daca persoane sau autovehicule au fost detectate sau efectiv nivelul de poluare fonica este alarmant.

**Testare ipoteze**

Pentru a ajunge la solutia finala, diverse implementari si solutii vor fi testate si analizate din punct de vedere al corectitudinii si integritatii rezultatelor, prin anumite metrici de evaluare. Totodata, o sa fie analizat si felul in care interpretarea datelor ajuta in rezolvarea problemei alese.

**Implementarea soluției propuse**

Pentru a identifica o solutie care sa satisfaca cerintele este nevoie de o tehnica de invatare automata care odata implementata si ulterior integrata in solutia finala, sa ofere o analiza si o sa putem masura o acuratele a clasificarii zgomotelor de tip ”persoana” si ”autovehicul”. Solutia propuse, pentru a fi accesibila pentru testare si utilizare va fi implementata ca o aplicatie Android ce o sa ofere o interfata ce va fi actualizata in timp real cu datele primite ca input, dar si cu datele celorlalti utilizatori, actualizate din cloud.

**Testare si evaluarea solutiei propuse**

Pentru a intepreta nivelul de performanta al solutiei alese, o sa se efectueze o analiza si o comparatie in acelasi timp cu alte solutii deja existente si a celei implementate si descrisa in acest document. Acest lucru o sa se realizeze cu ajutorul unor grafice ce vor indica prformanta obtinuta.

**Interpretabilitate**

Solutia va justifica alegerea facuta prin prezentarea nivelului de acuratete al detectiei cu ajutorul unor progress bar-uri, astfel se poate oferi o vizualizare a comportamentului modelului asupra datelor.

**2.3 Provocari**

**Senzori necalibrați**

Senzorii necalibrați reprezinta o problmea semnificativa atunci cand vine vorba de masurarea poluarii fonice. Calibrarea este un proces de ajustare si aliniere a seonzorilor pentru a asigura masuratori facute cu acuratete si de incredere. Inacuratetea in masuratori poate duce la rezultate eronate si a oferi valori indepartate de adevar. Acest lucru este firesc sa se intample deoarece dispozitivele mobile au senzorii calibrati astfel incat sa functioneze cu in conformitate cu sistemul de operare al telefonului si astfel pot aparea mici discrepante intre valoarea calculata si valoarea reala.

**Contextualizarea detecției**

De multe ori, pentru a ne da seama ca o mașină este in apropiere, nu ne putem baza doar pe sunetul produs de accelerația acesteia. De aceea trebuie luat in considerare si frânările, claxonatul, sunetul motorului. Astfel detecția va avea mai multa acuratețe.

**2.4 Cerinte functionale**

Principala functionalitate pe care aplicatia trebuie sa o indeplineasca este detectia zgomotelor si calcularea numarului de decibeli pentru a indica poluarea fonica, detectia persoanelor si a masinilor, stocarea si preluarea datelor in timp real din cloud pentru a oferi o privire de ansamblu asupra unei zone intregi.

**2.5 Cerinte non-functionale**

2.5.1 Scalabilitatea

Aplicatia trebuie sa fie scalabila, astfel ar trebui sa functioneze chiar daca pe parcurs o sa se adauge noi functionalitati, seturile de date o sa se schimbe sau arhitectura proiectului va suferi schimbari.

2.5.2 Performanta

Sistemul trebuie sa stocheze datele in timp real si in acelasi timp atunci cand se produce o schimbare sa actualizeze interfata pentru a oferi informatii relevante. Totodata, dorim performanta din punct de vedere al rezultatelor sa fie ridicata si ca acestea sa poata fi interpretate ulterior.

# Studiu bibliografic

In acest capitol sunt prezentate o parte din solutiile existente in domeniul detectiei poluarii fonice si solutii existente pentru detectia zgomotelor su reglementari cu privire la masurarea poluarii fonice.

**3.1 Metodologia măsurării zgomotelor**

Indicatorii de zgomot sunt folositi sa determine nivelul de zgomot dintr-un mediu. In general, indicatorii de sunet sunt descriptori ce indica valoarea zgomotului in decibeli. Valoarea unui indicator de zgomot de mediu este deteminata prin implementarea unui sistem de masurari si de estimare, dupa anumite standarde.

Inainte de a incepe procesul de masurare este necesar sa definim obiectivul masuratorii, a alinia obiectivele si implementarea, definirea surselor de zgomot si intervalelor de timp masuratorilor.

Calcularea anumitor indici de zgomot dintr-un anumit context, precum traficul, pot fi dificili de implementat deoarece intr-un mediu deschis se afla mai multi factori ce influenteaza masuratoarea si diverse surse de zgomot. De altfel, atat din punct de vedere științific si profesional, se folosesc indicatori de zgomot de mediu [1]. Cel mai popular indicator este Lden, care este format din diferentele de sunet din timpul zilei Lday, seara Levening si noapte Lnight. Acesta permite compararea mai multor infrastructuri si poate fi reprezentat prin harti de zgomot [1]. Masurarea sunetului se face 24h pe zi calendaristica, ce este impartite in trei intervale de referinta:

* Ziua durează 12h (06:00 – 18:00) Lday
* Seara durează 4h (18:00 – 22:00) Levening
* Noaptea durează 8h (22:00 – 06:00) Lnight

Nivelul zgomotului Lden pentru zi-seara-noapte este calculat in decibeli dupa urmatoarea ecuatie [2]:

Lday = 10log(12⋅100.1⋅Lday + 4⋅100.1⋅ (Levening + 5) + 8⋅100.1⋅(Lnight+10))

Unde Lday, Levening si Lnight sunt media ponderata A pe termen lung a nivelelor de sunet, determinate pentru toate zilele, serile si noptile de-a lungul unui an. Media ponderata A este filtrul standard de pondere in frecventa pentru nivelurile de sunet masurate cu ajutorul unor instrumente, utilizata de obicei in multe standard nationale si international [3]. Aceasta acopera intervale audio de la 20 la 20kHz si coreleaza cu intensitatea receptata de catre urechea umana. Unitatea de masura pentru masuratori este dB(A).

3.1.1 Analiza caracteristicilor sunetului

Pe langa calculul indicatorilor de sunet, caractertisticile acestuia pot fi studiata analizandu-i spectrele de frecventa. Transformata Fourier Rapida (FFT) s-a dovedit a fi un instrument foarte puternic in calcularea caracteristicilor sunetului [4]. Unul din beneficile transformarii semnalul in domeniul frecventelor este ca volumul de date este redus si de asemenea datele pot fi transferate printr-un mediu de comunicare mai repede; totusi, datorita necesitatilor computationale ridicate, analiza spectrala necesita mai mult timp de procesare si consum mai mare de energie [5].

**3.2 Analiza solutiilor deja implementate**

In ziua de azi, zgomotul produs de trafic este de obicei masurat cu dispozitive specializate. Mai mult decat atat, aceste dispozitive pot sa ofere date aditionale, iar din aceasta cauza este nevoie de un spatiu de stocare foarte mare pentru volumul de date obtinut. Recentele avansari in inginerie, in particular tehnologiile de tipul big data, au eficientizat felul in care se stocheaza volume mari de date si ofera analize avansate in timp real. Odata cu cresterea folosirii dispozitivelor mobile, modalitati de colectare de date mai facile au aparut. Tehnologia crowdsensing ofera colectarea informatilor referitoare la mediu cu ajutorul sensorilor implementati sau alte dispozitive si sa transmita datele receptionate catre centrele de colectare [6-9].

Metodele traditionale de masurat zgomotul sunt scumpe si de obicei se aplica in masuratori oficiale facute de catre agentii specializate [10]. In contrast cu aceste tehnici sunt studii ce cauta sa exploreze potentialul pe care il ofera dispozitivele mobile pentru masurari si analiza traficului [10-13]. Majoritatea acestor studii se bazeaza pe aplicatii mobile precum EarPhone, NoiseSpy, NoiseTube, NoiseBatle si NoiseMap. Undele din aceste aplicatii sunt distribuite la nivel comercial disponibile in magazinul de aplicatii predestinat dispozitivelor de ruleaza Android si iOS, in timp ce altele au fost dezvoltate pentru interes de cercetare.

3.2.1 EarPhone

In acest studiu [10] autorii descriu ca dezvoltarea aplicatiei EarPhone ce cartografiaza zgomotul este un sistem bazat pe crowdsensing. Sistemul EarPhone consta intr-o aplicatie mobila si un server central. Sistemul include un software pentru procesarea de semnale si masurarea sunetului de catre telefonul mobil, precum si un software pentru reconstruirea semnalului pe serverul central. Testarea sistemului a constat in masurarea nivelul de zgomot la reconstruirea unui drum principal din Brisbane. Masuratoarea a fost efectuta timp de o saptamana, in intervalele orare de 8-9h si 14-15h.Opt masuratori au fost efectuate in fiecare ora, iar in fiecare ora participanti la test mergeau timp de cinci minute si colectau date. Autorii acestui studiu se astepta ca sistemul lor sa reduca semnificativ costurile de operare in comparatie cu utilizarea dispozitivelor de inregistrat si cartografiat zgomote traditionale.

3.2.2 NoiseSpy

Sistemul de masuratori NoiseSpy ofera o vizualizare a colectilor inregistrate in timp real in timp ce se exploreaza diferite parti ale orasului [11]. Sistemul de masurare al zgomotului foloseste microfonul incorporat al telefonului si GPS-ul pentru a determina locatia exacta a masuratorii. Testarea sistemului a constat in amplasarea aplicatiei pe bicicletele curierilor. Pe parcursul rutelor acestora, sistemul masura nivelul de zgomot si il atribuia locatiei oferita de GPS [11]. Un concept similar pentru masurare zgomotului a fost prezentat in studiul [12] testand aplicatia crowdsensing NoiseTube. Spre deosebire de alte aplicatii, NoiseTube ofera abilitatea de a atasa diverse etichete la anumite inregistrari, distribuind datele direct din aplicatie, aceasta fiind disponibila pe mai multe platforme precum Android si iOS [14]. Pe langa aplicatia mobile, NoiseTube ofera de asemenea o platforma web ce permite utilizatorilor sa cerceteze, vizualizele si sa cauta datele. NoiseTube este aplicatia ce recalibreaza locatia pentru a reduce erorile de pozitionare [15].

In figura 3.2.1 este prezentata arhitectura sistemului NoiseSpy in conformitate cu lucrarea [11].

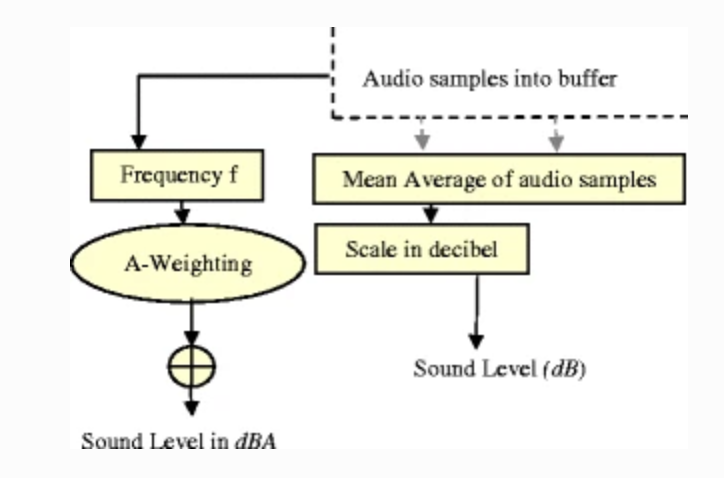


Figura 3.2.1: Arhitectura logica a sistemului NoiseSpy

Autorii studiului [16] au descporit o metoda inovativa de a anima cetatenii in timp ce colectau si distribuiau date cu privire la zgomot. Folosind proiectul open-source

NoiseDroid pentru a masura zgomotul, ce necesita utilizatorii sa se miste in tot orasul pentru a efectua masuratori ale zgomotului. In figura 3.2 este prezentata o astfel de abordare. Orasul este impartit in mai multe parti, iar utilizatorul care „castiga” trebuie sa faca mai multe masuratori decat ceilalalti participanti. Pe langa numerul de masuratori, algoritmul also ia in calcul si calitatea sunetului masurat.

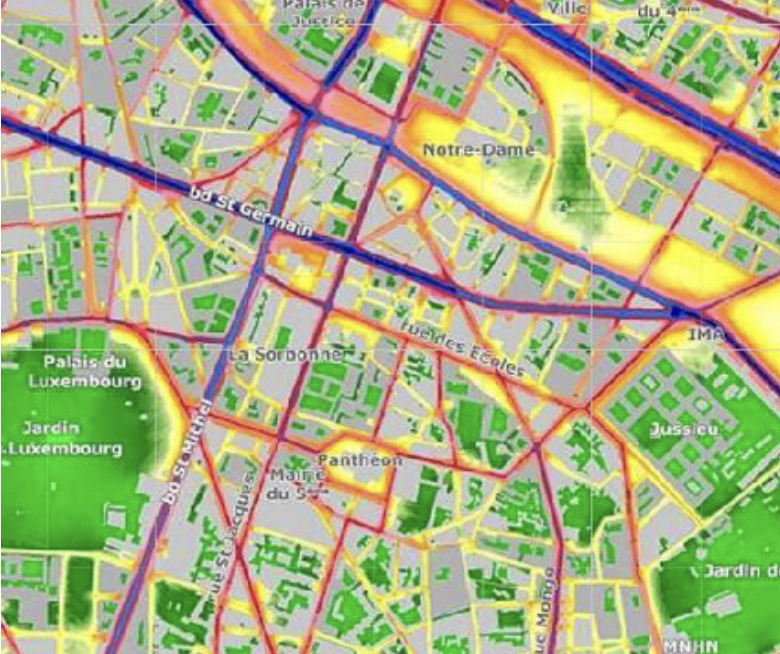


Figura 3.2.2: harta de zgomot a orasului Paris generata folosind propagarea unui model si anumite masuratori efectuate intr-un numar limitat de locatii. Zonele colorate in verde sunt zone cu zgomot redus, tin timp ce zonele zgomotoase sunt colorate in albastru/violet. Zonele gri reprezinta zone fara informatii disponibile (ex. in cladiri)[14].

Cercetatorii de la Universitatea Tehnica din Darmstadt au implementat aplicatie NoiseMap, ce colecteaza date si trimite in platforma web da\_sense [13]. Platforma permite utilizatorilor sa vizualizeze datele colectate, sa genereze grafice si harti de zgomot. Intr-un studiu ce a avut loc in 2012, cercetatorii au creat o harta de zgomot pentru orasul Frankfurt, luand in calcul doar șoselele pe unde trec mai mult de 6 milioane de masini anual. Spre deosebire de alte proiecte, datele sunt disponibile pe o platforma web publica sau printr-un JavaScript API.

Bazat pe analiza facuta anterior, putem concluziona ca cele mai multe sisteme de masurare a zgmotoului utilizate in orasele smart folosesc: 1) aplicatii mobile ce folosesc microfonul pentru a colecta date legate zgomot si un sistem GPS pentru locatia in care masuratoarea a fost efectuta; 2) platforme web pentru vizualizarea datelor legate de zgomot si locatie. Totodata, nu se mentioneaza un plan pentru stocarea pe termen lung si a managementului calupului mare de date colectat sau capabilitatea de a face analize complexe in timp real. Avantajele implementarii sistmeului prezentat in lucrarea aceasta spre deosebire de cele mentionate anterior sunt: 1) dezvoltarea unui sistem ce permite stocarea datelor intr-o baza de date ce se afla in cloud si este accesibila oricand; 2) datele colectate pentru analiza sunt interpretate pentru a distinge masinile si persoanele de alte surse de zgomot din exterior, ceea ce permite o analiza mai sofisticata a zgomotului; 3) analiza este bazata pe datele masurate disponibila in timp real.

**3.3 Acuratetea si precizia masuratorilor de zgomot**

Acuratetea dispozitivelor pentru masurat zgomote, in termen de apropiere a masuratorilor cu valorile fizice, realce, a fost considerata in studii din mai multe puncte de vedere. In contextul de crowdsensing, acuratetea folosirii telefoanelor mobile pentru masurarea zgomotului a stârnit interes. Vorbind despre comparatia directa a capabilitatii telefoanelor mobile de a fi calibrate pentru a exprima metrici de masurare a sunetului, aceasta a generat un nivel de acuratete acceptabil, conducand astfel la concluzia ca crowdsensing-ul este o metoda acceptabila de masurare a zgomotului generat de trafic [17]. In general, acuratetea masurarii sunetului cu ajutorul telefoanelor mobile depinde de caracteristicile sistemului de operare, producatorului, modelului si conditia din punct de vedere al functionarii a dispozitivului. [18, 19]. De asemenea, acuratetea depoinde si de microfonul inocrporat in telefonul mobil. Rezultatele arata faptul ca intervalul in care se incadreaza inregistrarile de pe un telefon mobil sunte intre 50 si 90 dB(A), ceea ce corespunde cu nivele de zgomot obisnuite inregistrate din asezarile urbane [19].

Acuratetea masuratorilor poate fi imbunatatita cu ajutorul unei calibrari telefonului mobil. Calibrarea se efectueaza de obicei cu sonometre (SLMs). Datorita faptului mentionat anterior si anume ca diferentele individuale dintre dispozitivele mobile [18], nu este cu adevarat posibil sa existe calibrare la telefoanele mobile ale participantilor in crowdsensing; in acest caz ar trebui aplicate alte abordari [18, 20, 21]. Din analiza sutdiilor nu a rezultat nicio solutie ce este uniforma si general aplicabila pentru calibrare, nici macar pentru modele identice de telefoane mobile [22]. O posibila solutie pentru aceasta problema ar presupune ca fabricantii de dispozitive sa publice instructiunile tehnice cu privire la calibrarea propriilor dispozitive in conformitate cu standardele [23].

Pe langa acuratete, calitatea masuratorilor depinde si de precizie, de exemplu cat de apropiate sunt efectuate masuratorile, Precizia depinde de configuratia locatiei, pozitia dispozitivelor mobile in raport cu obstacolele fizice, orientarea dispozitivelor mobile, expunerea microfonului in raport cu sursele de zgomot, etc. [19]. Mai mult decat atat, precizia depinde si de miscarea participantilor [24]. In momentul in care mai multi participanti se afla in experimentul de crowdsensing, precizia poate fi imbunatatita urmarind protcoale de masurare stricte [23].

In contextul colectarii datelor prin crowdsensing, s-ar putea observa o mai mica acuratete si precizie in momentul in care sunt putine dispozitive in zona, lucru ce ar putea fi compensat printr-un numar ridicat de masuratori efectuat de un numar mai mare de dispozitive [25]. O fiabilitate mai mare a sumarizarii masuratorile este obtinuta combinand datele masute de mai multe device-uri si aplicand diverse metode statistice si tehnici pentru filtrarea datelor, explorare si analiza [26].

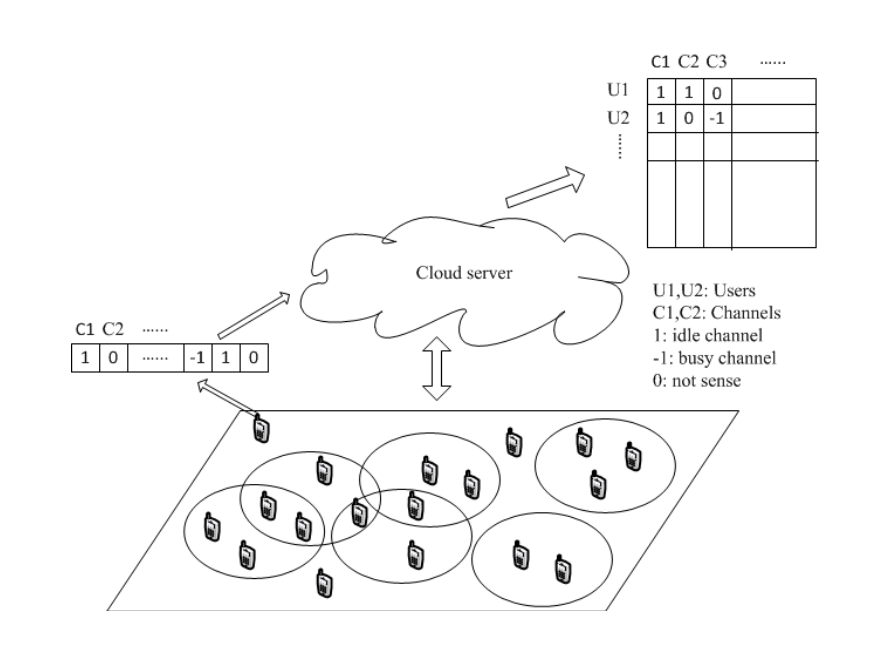


Figura 3.3.1: Descoperirea spectrului bazata pe fuziunea datelor provenite de la dispozitive mobile.

**3.4 Reglementari cu privire la măsurarea zgomotelor**

Uniunea Europeana este considerata ca fiind lider in domeniul masurarii zgomotelor [27]. Statele membre efectueaza cartografieri ale zgomotelor de multi ani, datorita recomandarilor din cadrul directiver UE 2002-49-EC [28]. Directive a incurajat avansul in metodele de masurare a zgomotului si in crearea de harti de zgomot pentru mai multe orase importante [29]. Cu privire la poluarea fonica, UE are diferite legi prevazute pentru traficul la sol, traficul aerian, cat si pentru sursele de zgomot industrial. In zonele urbane, doar soselele principale, liniile principale de cale ferata si aeroporturile principale sunt luate in considerare. Spre deosebire de Europa, unde cartografierea zgomotului este obligatorie, Statele Unite ale Americii si Japonia au legi ce dateaza din anii 1970, dar cartografierea zgmotoului este optionala. In dezvoltarea tarilor, de obicei este o lipsa la legislaturii pentru masuratori, evaluari si control al nivelelor de zgomot [30].

Conform lucrarii [31] se observa raspunsul populatiei mai multor orase importante in care se exprima nivelul de satisfacere cu nivelul de zgomot local.

A picture containing text, map, atlas

Description automatically generated

Figura 3.4.1: Proportia de persoane satisfacuta cu nivelul de zgomot in orasele unde locuiesc, 2015

# Analiză și fundamentare Teoretică

In acest capitol umreaza sa fie prezentata solutia aleasa, dupa care o sa fie descrise in detaliu conceptele teoretice si algoritmii utilizati pentru implementare.

**4.1 Solutia propusa**

Solutia propusa presupune utilizarea unui model pre-antrenat, integrat intr-o aplicatie Android, alaturi de senzorii incorporati ai telefonului mobil, iar acestea insumate rezultand intr-un sistem capabil sa recunoasca persoanele si masinile din apropiere, bazat pe datele de intrare de la microfonul telefonului, sa calculeze poluarea fonica in timp real sa amplaseze pe harta zonele afectate de poluare fonica indiferent daca utilizatorul in miscare sau nu. Datele receptionate vor fi transmise in cloud, iar ceilalti utilizatori pot beneficia de ele in timp real. Pentru realizarea sistemului s-au efectuat urmatoarele procese prezentate in figura 4.1.1, in care datele provenite de la microfonul telefonului sunt calibrate, iar ulterior sunt trimise catre modelul propriu-zis.

A picture containing text, screenshot, diagram, font

Description automatically generated

Figura 4.1.1 Diagrama generala a sistemului

Modelul a fost antrenat pe un set de date ce este bazat pe sunetul din diverse clip-uri video, din care au fost extrase segmente de 10 secunde, etichetate de oameni si impartite in ontologii. O ontologie este o colectie de sunete organizate intr-o ierarhie, iar aceasca acopera o arie larga din sunetele pe care le receptam in fiecare zi precum, sunete produse de oameni si animale, sunete din natura si din mediul inconjurator, muzica, trafic, etc. Pentru antrenarea modelului in ceea ce priveste detectia vehiculelor au fost folosita urmatoarea ponderea de material video.

A screenshot of a data

Description automatically generated with low confidence

Figura 4.1.2: Antrenarea modelului pentru detecția vehiculelor

**4.2 Implementarea TensorFlow in Android**

TensorFlow este un framework open-source de machine-learning, ce ofera diverse librarii pentru clasificare audio. Acesta poate fi modelat in functie de cerintele functionale ale proiectului. Pentru clasificarea audio urmatoarii pasi de preprocesare si procesare de date sunt necesari.

4.2.1 Preprocesarea datelor audio

TensorFlow ofera diverse functii si libarrii pentru a decodifica diverse formate audio precum mp3 sau WAV, iar mai apoi se eșantionează pentru a avea acelasi eșantion cu modelul. Formele de unda se normalizeaza in intervalul -1 si 1, pentru a se asigura nivele constante de amplitudine. De obicei acest procedeu se efecuteaza impartind datele audio la valoarea lor maxim absoluta. Acestea sunt impartite apoi in secvente mai mici de cateva milisecunde si se calculeaza caracteristicile relevante din fiecare secventa audio. Cele mai comune caracteristici includ: Mel-frequency cepstral coefficients(MFCC)[[5]](#footnote-5) si spectograme ce reprezinta magnitudinea sau putere spectrala a semnalului.

4.2.2 Augmentarea datelor si procesarea

Tehnici pentru augmentarea datelor pot fi aplicate pentru a creste artificial dimensiunea si diversitatea data setului de invatare. TensorFlow include functionalitati pentru implementarea tehnici ca si parte a preprocesarii. Arhitectura clasificarii audio este efectuate de API-uri de nivel inalt precum Keras[[6]](#footnote-6) sau de nivel scazut precum tf.nn[[7]](#footnote-7). De obicei arhitectura acestor modele este de tipul CNNs (retele neuronale convolutionale) sau RNNS (retele neuronale recurente). Antrenarea acestora se efectueaza cu datele preprocesate si codificate sub forma de date de intrare. Parametrii modelelor se pot optimiza folosind algoritmi specifici pentru acest lucru.

4.2.3 Integrarea TensforFlow in solutia propusa

Pentru integrarea TensorFlow in cadrul aplicatiei Android creata a fost nevoie de folosirea librariei TensorFlow LiteTask Library[[8]](#footnote-8). Urmatorul pas consta in filtrarea output-ului venit pe baza detectiei, afisarea acestuia si actualizarea datelor pe baza a ceea ce se doreste. Pentru a face lucrul acesta a fost nevoie sa implementez un Adapter[[9]](#footnote-9). In programarea in Android, un adapter este de fapt un canal de comunicare intre interfata grafica si baza de date. Un view de tip adapter se asigura ca fiecare element de pe interfata grafica corespunde unui element din baza de date, formand o legatura intre ele, astfel incat orice actiune atat de pe frontend cat si de pe backend sa se propage in ambele zone. Astfel in momentul in care un element este recunoscut pe baza sunetului emis (ex. un autovehicul), pe interfata grafica va aparea informatia respectiva, va si scrisa automat si in baza de date, iar eticheta respectiva va disparea in momentul in care nu este nimic de detectat.

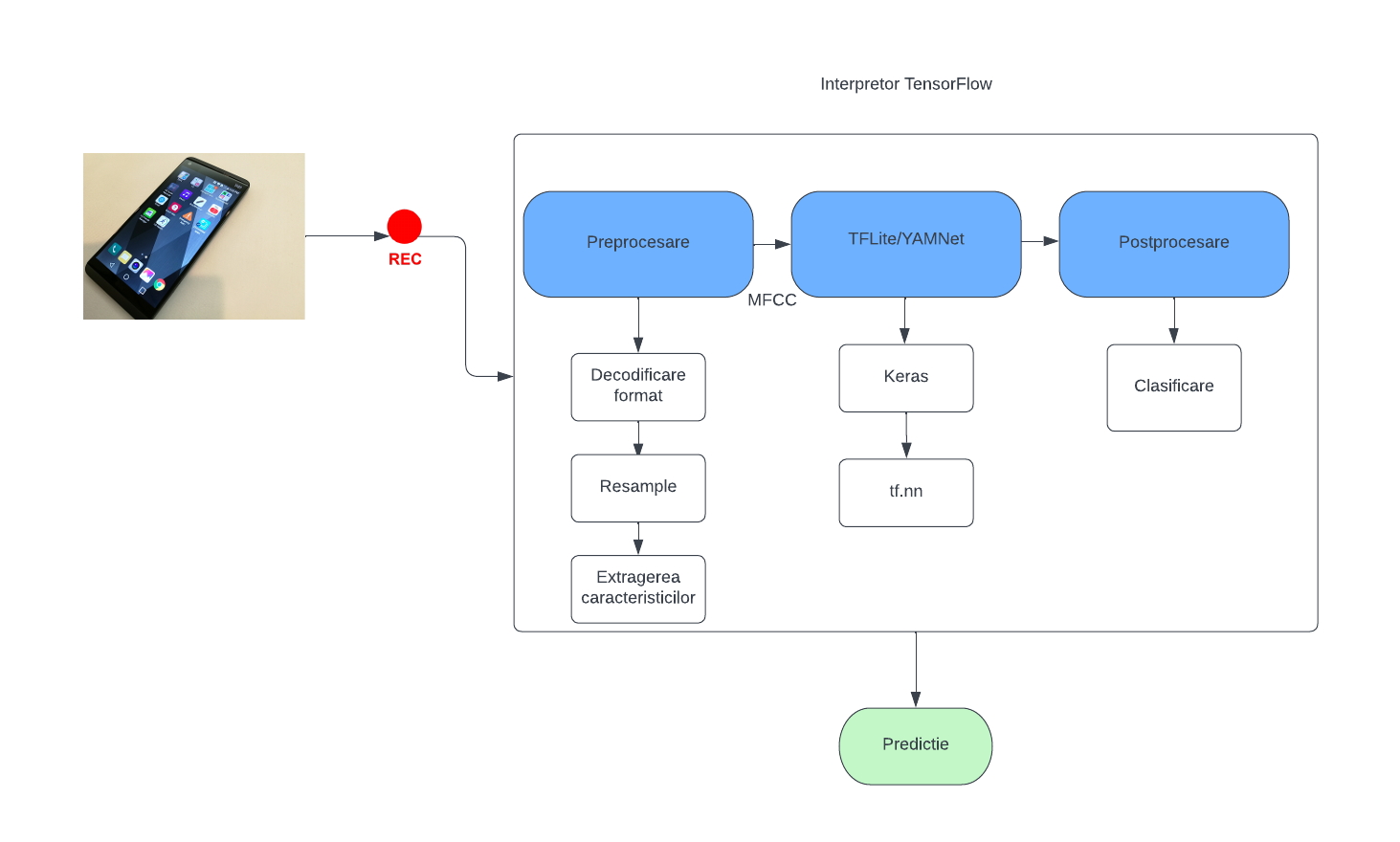


Figura 4.2.1: Conexiunea TensorFlow cu Android

**4.3 Localizarea in timp real in cadrul aplicației**

Locația utilizatorilor, cu acordul acestora, este o caracteristica a solutiei propuse deoarece aceasta presupune amplasarea cu exactitate a zonelor de interes si a informațiilor necesare ce corespund fiecărei zone.

4.3.1 Cererea permisiunilor pentru locație

Pentru a accesa locatia utlilizatorilor, aplicatia Android trebuie sa ceara permisiunile necesare. Permisiunea ”ACCESS\_FINE\_LOCATION” este necesara, deoarece aceasta permite aplicatiei sa acceseze datele GPS exacte. Pentru a obtine aceasta permisiune urmatorii pasi trebuie urmati: 1) in cadrul fisierului AndroidManifest, unde se gasesc detalii cu privire la versiunile folosite in cadrul aplicatiei, se mai adauga o sectiune (<uses-permission>) in care se mentioneaza ca aplicatia foloseste permisiunile cerute; 2) cererea permisiunilor in care aplicatia ruleaza, astefel in cazul in care acestea sunt acordate, amplasarea pe harta se va face conform cerintei; 3) verificarea faptului ca sistemul de localizare din cadrul sistemului este activ, iar daca acesta nu este, cererea permisiunii pentru activare din cadrul aplicatiei automat.

4.3.2 Servicii de localizare si furnizori

In mediul Android sunt numeroase servicii de localizare si furnizori pentru a obtine informatii cu privire la locatie. Cei mai intalniti sunt cei de tipul GPS, locatia bazata pe retea si furnizori de locatie fuzionati. Furnizorii de locatie fuzionati combina datele din mai multe surse pentru a asigura acuratete si eficienta in actualizarea locatiei. Astfel, pentru a integra servicii de localizare am realizat conexiunea folosind serviciile Google Play API[[10]](#footnote-10), astfel formandu-se legatura cu furnizorul de locatie fuzionat. Ulterior a setat un LocationListener[[11]](#footnote-11) pentru a obtine actualizari de la furnizor. Iar pentru a receptiona schimbarile am implementat apelul onLocationChanged()11 pentru a primi actualizari ale locatiei, mai exact latitudinea si longitudinea.

4.3.3 Integrarea localizarii in timp real

Pentru a asigura localizarea in timp real in aplicatie, actualizarile cu privire la locatia utilizatorului trebuie integrate si utilizate. Pentru a obtine acest lucru trebuie integrata o harta; acest lucru este posibil prin integrarea cu Google Maps API[[12]](#footnote-12) pentru a putea afis locatiile utilizatorilor si a actualiza in timp real actiunile acestora. Pentru rularea in fundal a acestui lucru este necesara cererea de permisiuni pentru obtinerea locatiei chiar daca utilizatorul nu se afla in cadrul aplicatiei. Toate aceste date mentionate anterior trebuie stocate intr-un loc centralizat, iar pentru asta am ales Firebase, ce va fi prezentat in capitolele urmatoare.

A diagram of a flowchart

Description automatically generated with low confidence

Figura 4.3.1: Diagrama ce descrie declararea si cererea permisiunilor la rulare Android.

**4.4 Calcularea numarului de decibeli cu ajutorul telefonului**

Pentru calcularea numarului de decibeli in solutia propusa, trebuie obținută permisiunea pentru acces la microfonul telefonului apoi detectia sunetului in SPL (nivelul de presiune al sunetului) si convertit in dB(A).

4.4.1 Preprocesarea datelor audio obținute

Pentru a obtine datele audio prin microfonul incorporat al dispozitivului o sa ne folosim de o caracteristica incorporata in Android si anume AudioManager. Acesta ofera acces la datela audio ca si flux, lucru ce va fi folositor pentru procesarea ulterioara. Odata ce datele sunt obtinute, trebuie facuta o conversie deoarece initial, datele audio brute sunt reprezentate ca si forme de unda, dar pentru a calcula SPL trebuie sa fie convertite intr-o reprezentare numerica. Dupa ce componentele de frecventa au fost extrase, valoarea SPL poate fi calculata astfel:

SPL(dB) = 20 \* log10(Presiune / Referinta).

Presiunea se refera la amplitudinea presiunii sonore, iar referinta pentru nivelul presiunii sunetului este de obicei 20 μPa.

4.4.2 Aplicarea filtrelor de pondere si conversia in dB(A)

Termenul dB(A) este o unitate de masura pentru sunet ce ia in considerare variatia sensibilității sistemului de auz al oamenilor. Aplicarea filterlor de pondere A, reprezinta frecvente de pondere aplicate pe valorile nivelului presiunii sunetului (SPL). Pentru a se face conversia din SPL in dB(A), valoare SPL trebuie inmultita cu coeficientii filterlor de ponderea A, astfel valoarea obtinuta este aceeasi cu nivelul perceput de urechea umana. Aceasta valoare odata calculata va fi stocata in baza de date din cloud, va fi actualizata mereu si va fi disponibila pe interfata grafica pentru utilizatori.

**4.5 Integrarea Firebase**

Firebase este o platforma dedicata dezvoltarii aplicatiilor mobile si web ce ofera o gama larga de servicii si resurse. A fost preluata de cei de la Google in 2014 si de atunci a devenit una din cele mai popularea variante pentru construirea de aplicatii scalabile. La baza, Firebase ofera o infrastructura de backend bazata pe cloud ce simplifica mai multe sarcini ce trebuie indeplinite. Acesta ofera o suita de beneficii ce pot fi folosite la implementare precum stocare de date in timp real, autentificare, servicii de administrare si multe altele.

4.5.1 Firebase Authentication

In orice sistem, la fel ca si in cel propus, trebuie sa existe o parte de autentificare. Pentru a asigura acest lucru in solutia propusa, am folosit Firebase Authentication, care este un sistem securizat de permite autentificarea utilizatorilor si autorizarea acestora. Firebase Authentication suporta autentificarea prin mai multe moduri precum: autentificarea cu numarul de telefon, autentificarea prin diverse retele sociale precum Google, GitHub, Facebook si Twitter, dar si prin email si parola.

Pentru a integra acest serviciu am folosit un SDK[[13]](#footnote-13) oferit de cei de la Firebase[[14]](#footnote-14) ce permite incapsularea intr-o aplicatie Android. Prin comunicarea efectuata in cadrul aplicatiei utilizatorii pot efectua operatii de inregistrare si logare securizate.



Figura 4.5.1: Modalitatile de logare folosind Firebase Authentication

4.5.2 Firebase RealTime Database

Firebase RealTime Database este o baza de date in cloud de tipul NoSQL[[15]](#footnote-15) oferita de Firebase. Aceasta ofera sincronizare in timp real, posibilitate de stocare de date, oferind astfel optiunea de a construi aplicatii ce pot sa colecteze date si sa se actualizeze in timp real.

Acest sistem, spre deosebire de cele clasice in care structura era sub forma tabelara, primeste si inmagazineaza datele sub forma de JSON, adica o forma arborescenta. Pentru integrarea acestei caracteristici cu solutia propusa a fost nevoie de incadrarea anumitor dependinte in fisierul build.gradle[[16]](#footnote-16). Astfel, datele cu privire la poluarea fonica provenite de la utilizatori pot fi receptionate, procesate si transmise catre alti utilizatori in timp real.

A picture containing text, diagram, plan, rectangle

Description automatically generated

Figura 4.5.2: Stocarea si actualizarea datelor in timp real folosind Firebase RealTime Database.

4.6.2 Firebase Functions

Firebase Functions reprezinta o platforma oferita de firebase ce permite rularea de cod pentru backend precum schimbari in baza de date, autentificarea utilizatorilor sau operatii pentru intretinerea infrastructurii.

In solutia propusa a fost necesara integrarea acestui modul pentru a putea prelua datele anonime ale fiecarui utilizator si anume locatia si datele cu privire la poluarea fonica. Utilizatorii sunt stocati dupa id-ul autogenerat de sistemul de autentificare, ceea ce nu pericliteaza informatiile cu caracter personal ale acestora.

Functiile au fost scrise in TypeScript, dupa care s-a facut deploy pe serverul dedicat oferit de Firebase. Dupa integrarea dependintelor necesare in proiect, functiile scrise au putu fi apelate.

**4.6 Arhitectura MVVM**

Arhitectura folosita in dezvoltarea proiectului este Model-View-ViewModel (MVVM). Aceasta este raspandita in dezvoltarea aplicatiilor android deoarece ofera o strucutra organizata in construirea aplicatiilor robuste si scalabile. MVVM separa perocuparile cu privire la integrarea datelor pe interfata grafica si procesarea proriu-zisa a lor, rezultand astfel o separare a responsabilitatilor in implementare.

4.6.1 Model

Modelul reprezinta datele si partea de bussiness logic[[17]](#footnote-17) ale aplicatiei. Acesta incapsuleaza sursele de date precum bazele de date locale, API-uri sau baze de date aflate in cloud. Modelul este responsabil pentru aducerea, manipularea si persistenta datelor. Nu este dependent de View si ViewModel. In solutia propusa, modelul se preocupa de colectarea datelor de la senzorii telefonului, procesarea acestora si adaugarea in baza de date.

4.6.2 View

View-ul reprezinta componentele interfetei grafice, in Android definite ca si Activitati[[18]](#footnote-18) si Fragmente[[19]](#footnote-19). Acesta este responsabil de afisarea datelor catre utilizatori si sa se ocupe de interactiunile cu acestia. View-ul nu trebuie sa contina elemente de bussiness logic sau sa mentina starea in care se afla aplicatia.

4.6.3 ViewModel

ViewModel-ul functioneaza ca un canal de comunicare intre Model si View. Acesta expunde datele si operatiile de care View-ul are nevoie. ViewModel-ul primeste date de la Model, le formateaza si le pregateste pentru prezentare. El de asemeneaza mediaza si interactiunile pe care utilizatorul le efectueaza din View si le comunica mai departe Model-lului pentru a se ocupa de operatiile necesare. Acesta nu trebuie sa mentina starea in care se afla View-ul, asigurand separarea intre interfata grafica si bussiness logic.

# Proiectare de detaliu și implementare

Împreună cu capitolul precedent și cel următor reprezintă aproximativ 70% din total.

Scopul acestui capitol este să documenteze aplicația dezvoltată în așa fel încât dezvoltarea și întreținerea ulterioară să fie posibilă. Cititorul trebuie să poată identifica funcțiile principale ale aplicației din ceea ce este scris aici.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* schema generală aplicației,
* descriere a fiecărei componente implementate, la nivel de modul,
* diagrame de clase, clase importante și metode ale claselor importante.

# Testare și validare

Acest capitol, împreună cu cele două care îl preced, va reprezenta aproximativ 70% din lucrare.

# Manual de instalare si utilizare

În secțiunea de Instalare trebuie să detaliați resursele software și hardware necesare pentru instalarea și rularea aplicației, precum și să descrieți pas cu pas procesul de instalare. Instalarea aplicației trebuie să se poată face folosind instrucțiunile date aici.

Utilizarea aplicației trebuie descrisă din punctul de vedere al utilizatorului, fără a menționa aspecte tehnice interne. Folosiți capturi ale ecranului și explicați pas cu pas interacțiunea cu persoana care execută instalarea. Folosind acest manual, o persoană ar trebui să poată instala și utiliza produsul vostru.

Minimum 1 pagină, până la 5 pagini

# Concluzii

Acest capitol va ocupa 1-2 pagini.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* un rezumat al contribuțiilor voastre
* o analiză critică a rezultatelor obținute
* o descriere a posibilelor dezvoltări și îmbunătățiri ulterioare

# Bibliografie

|  |  |
| --- | --- |
| [1] | C. Pronello, C. Camusso, A review of transport noise indicators, Transp. Rev. 32 (2012) 599–628, <https://doi.org/10.1080/01441647.2012.706332> . |

[2] [Mark Brink, Beat Schäffer, Reto Pieren, Jean Marc Wunderli Conversion between noise exposure indicators Leq24h, LDay, LEvening, LNight, Ldn and Lden: Principles and practical guidance](https://www.sciencedirect.com/science/article/abs/pii/S1438463917304819?casa_token=iBVDSvkkkwYAAAAA:kdfA_X9jgL7ASrfCrIKo8fSq3tD1gIXB8esFsxIb7CzwOfaKnlvc85db4FivfMfAP9rmW4ic)

**[3]** A.-M.O. Mohamed, E.K. Paleologos, F.M. Howari, Noise pollution and its impact on human health and the environment, Pollut. Assess. Sustain. Pract. Appl. Sci. Eng. (2021) 975–1026, <https://doi.org/10.1016/b978-0-12-809582-9.00019-0> . Elsevier.

**[4]** J. Púčik, P. Kubinec and O. Ondřáček, "FFT with modified frequency scale for audio signal analysis," 2014 24th International Conference Radioelektronika, Bratislava, Slovakia, 2014, pp. 1-4, doi: 10.1109/Radioelek.2014.6828482.

[5] [K.R. Rao, D.N. Kim, J.J. Hwang, Fast Fourier Transform - Algorithms and Applications, Springer, 2010.](https://link.springer.com/book/10.1007/978-1-4020-6629-0)

[6] L. Sanchez, L. Munoz, ˜ J.A. Galache, P. Sotres, J.R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, D. Pfisterer, SmartSantander: IoT experimentation over a smart city testbed, Comput. Netw. 61 (2014) 217–238, <https://doi.org/10.1016/j.bjp.2013.12.020>.

[7] M. Zappatore, A. Longo, M.A. Bochicchio, Using mobile crowd sensing for noise monitoring in smart cities, 2016 Int. Multidiscip. Conf. Comput. Energy Sci. Split. 2016 (2016), <https://doi.org/10.1109/SpliTech.2016.7555950>.

[8] G. Singh, D. Bansal, S. Sofat, A smartphone based technique to monitor driving behavior using DTW and crowdsensing, Pervasive Mob. Comput. 40 (2017) 56–70, <https://doi.org/10.1016/j.pmcj.2017.06.003>.

[9] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, R. Curtmola, Fostering participaction in smart cities: a geo-social crowdsensing platform, IEEE Commun. Mag. 51 (2013) 112–119, [https://doi.org/10.1109/ MCOM.2013.6525603](https://doi.org/10.1109/%20MCOM.2013.6525603).

[10] R.K. Rana, C.T. Chou, S.S. Kanhere, N. Bulusu, W. Hu, Ear-phone an end-to-end participatory urban noise mapping system, Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks - IPSN’ 10 (2010) 105–116, [https://doi.org/10.1145/ 1791212.1791226](https://doi.org/10.1145/%201791212.1791226) . ACM Press, New York, New York, USA

[11] E. Kanjo, NoiseSPY: a real-time mobile phone platform for urban noise monitoring and mapping, Mob. Netw. Appl. 15 (2010) 562–574, [https://doi.org/10.1007/ s11036-009-0217-y](https://doi.org/10.1007/%20s11036-009-0217-y).

[12] [STAPLES, E. EnviSensor: Environmental Data Collection via Participatory Sensor Networks Utilizing Mobile Devices. *Master’s thesis*, 2011.](https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=da64526acae44b036287bc62e51aac8b73fe4a10)

[13] [I. Schweizer, R. Artl, A. Schulz, F. Probst, M. Uhl Auser, NoiseMap -Real-time participatory noise maps, Second Int. Work. Sens. Appl. Mob. Phones (2011) 1–5.](https://hmr.biomedcentral.com/articles/10.1007/s10152-010-0194-3)

[14] N. Maisonneuve, M. Stevens, M.E. Niessen, L. Steels, NoiseTube: Measuring and Mapping Noise Pollution with Mobile Phones, Springer, Berlin, Heidelberg, 2009, pp. 215–228, <https://doi.org/10.1007/978-3-540-88351-7_16>.

[15] [M. Stevens, E.D. ’hondt, Crowdsourcing of pollution data using smartphones, Int. Conf. Entertain. Comput. (2012) 562–571.](https://www.researchgate.net/profile/Matthias-Stevens/publication/228911632_Crowdsourcing_of_Pollution_Data_using_Smartphones/links/00b4953a20b590bc8a000000/Crowdsourcing-of-Pollution-Data-using-Smartphones.pdf)

[16] I.G. Martí, L.E. Rodríguez, M. Benedito, S. Trilles, A. Beltr´ an, L. Díaz, J. Huerta, Mobile Application for Noise Pollution Monitoring through Gamification Techniques, Springer, Berlin, Heidelberg, 2012, pp. 562–571, [https://doi.org/ 10.1007/978-3-642-33542-6\_74](https://doi.org/%2010.1007/978-3-642-33542-6_74).

[17] S. Grubeˇsa, A. Petoˇsi´c, M. Suhanek, I. Đurek, Mobile crowdsensing accuracy for noise mapping in smart cities, Automatika 59 (2018) 286–293, [https://doi.org/ 10.1080/00051144.2018.1534927](https://doi.org/%2010.1080/00051144.2018.1534927).

[18] E. Murphy, E.A. King, Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise, Appl. Acoust. 106 (2016) 16–22, <https://doi.org/10.1016/j.apacoust.2015.12.012>.

[19] P. Aumond, C. Lavandier, C. Ribeiro, E.G. Boix, K. Kambona, E. D’Hondt, P. Delaitre, A study of the accuracy of mobile technology for measuring urban noise pollution in large scale participatory sensing campaigns, Appl. Acoust. 117 (2017) 219–226, <https://doi.org/10.1016/j.apacoust.2016.07.011>.

[20] J. Zuo, H. Xia, S. Liu, Y. Qiao, Mapping urban environmental noise using smartphones, Sensors 16 (2016) 1692, <https://doi.org/10.3390/s16101692>.

[21] E. D’Hondt, M. Stevens, A. Jacobs, Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring, Pervasive Mob. Comput. 9 (2013) 681–694, <https://doi.org/10.1016/j.pmcj.2012.09.002>.

[22] [C. Beyers, Calibration methodologies and the accuracy of acoustic data, in: C. Beyer (Ed.), INTER-NOISE NOISE-CON Congr. Conf. Proc., 2014, pp. 3104–3111. Institute of Noise Control Engineering, Melbourne](https://www.acoustics.asn.au/conference_proceedings/INTERNOISE2014/papers/p449.pdf).

[23] J.M. Barrigon ´ Morillas, D. Montes Gonzalez, ´ G. Rey Gozalo, A review of the measurement procedure of the ISO 1996 standard. Relationship with the European Noise Directive, Sci. Total Environ. 565 (2016) 595–606, [https://doi.org/10.1016/ j.scitotenv.2016.04.207](https://doi.org/10.1016/%20j.scitotenv.2016.04.207).

[24] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, ACM Trans. Sens. Netw. 6 (2010) 1–27, <https://doi.org/10.1145/1689239.1689243>.

[25] ] [X. Hao, L. Xu, N. Lane, X. Liu, T. Moscibroda, Density-aware compressive CrowdSensing, IPSN2017 (2017) 29–40](https://dl.acm.org/doi/10.1145/3055031.3055081).

[26] S. Lin, J. Zhang, L. Ying, Crowdsensing for Spectrum discovery: a waze-inspired design via smartphone sensing, IEEE/ACM Trans. Netw. 28 (2020) 750–763, <https://doi.org/10.1109/TNET.2020.2976927>.

[27] [Noise Nuisance and the Law, 2015](https://scholar.google.com/scholar?q=Noise%20Nuisance%20and%20the%20Law).

[28] [DIRECTIVE 2002/49/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 25 June 2002](https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:189:0012:0025:en:PDF).

[29] E. Murphy, E.A. King, Strategic environmental noise mapping: methodological issues concerning the implementation of the EU Environmental Noise Directive and their policy implications, Environ. Int. 36 (2010) 290–298, [https://doi.org/ 10.1016/J.ENVINT.2009.11.006](https://doi.org/%2010.1016/J.ENVINT.2009.11.006).

[30] [Community and regional noise mapping in the United States](https://www.researchgate.net/publication/279555007_Community_and_regional_noise_mapping_in_the_United_States).

[31] [Eurostat, 2016, Urban Europe — Statistics on cities, towns and suburbs — Eurostat, Cap. 6, Map 6.5](https://ec.europa.eu/eurostat/documents/3217494/7596823/KS-01-16-691-EN-N.pdf/0abf140c-ccc7-4a7f-b236-682effcde10f)

După ce ați inserat/actualizat bibliografia selectați întregul tabel și aplicați stilul *Biblio.*  Stilul *Normal* are indentare la începutul paragrafelor și, de aceea nu veți obține formatul ca mai sus fără acest pas.

# Anexa 1

…

Secțiuni de cod relevante

# Anexa 2

Alte informații relevante (demonstrații etc.)

…

# Anexa 3

Lucrări publicate (dacă există)

etc.

1. 1 [P. Ross, Top 11 technologies of the decade, Spectrum, IEEE 48 (2011) 27–63](https://ieeexplore.ieee.org/abstract/document/5676379) [↑](#footnote-ref-1)
2. [N. Eagle, A. Pentland, D. Lazer, Inferring friendship network structure by using mobile phone data, Proceedings of the National Academy of Sciences 106 (2009) 15274–15278.](https://www.pnas.org/doi/full/10.1073/pnas.0900282106) [↑](#footnote-ref-2)
3. [M.C. Gonzalez, C.A. Hidalgo, A.-L. Barabasi, Understanding individual human mobility patterns, Nature 453 (2008) 779–782](https://www.nature.com/articles/nature06958) [↑](#footnote-ref-3)
4. [G. Chittaranjan, J. Blom, D. Gatica-Perez, Mining large-scale smartphone data for personality studies, Personal and Ubiquitous Computing 17 (2013) 433–450.](https://link.springer.com/article/10.1007/s00779-011-0490-1) [↑](#footnote-ref-4)
5. <https://ismir2000.ismir.net/papers/logan_abs.pdf> [↑](#footnote-ref-5)
6. <https://keras.io/api/> [↑](#footnote-ref-6)
7. <https://www.tensorflow.org/api_docs/python/tf/nn> [↑](#footnote-ref-7)
8. <https://www.tensorflow.org/lite/inference_with_metadata/task_library/audio_classifier> [↑](#footnote-ref-8)
9. <https://developer.android.com/reference/android/widget/Adapter> [↑](#footnote-ref-9)
10. <https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices> [↑](#footnote-ref-10)
11. <https://developer.android.com/reference/android/location/LocationListener> [↑](#footnote-ref-11)
12. <https://developers.google.com/maps> [↑](#footnote-ref-12)
13. <https://en.wikipedia.org/wiki/Android_SDK> [↑](#footnote-ref-13)
14. <https://firebase.google.com/docs/auth> [↑](#footnote-ref-14)
15. <https://www.mongodb.com/nosql-explained> [↑](#footnote-ref-15)
16. <https://developer.android.com/build> [↑](#footnote-ref-16)
17. <https://en.wikipedia.org/wiki/Business_logic> [↑](#footnote-ref-17)
18. <https://developer.android.com/reference/android/app/Activity> [↑](#footnote-ref-18)
19. <https://developer.android.com/guide/fragments> [↑](#footnote-ref-19)