

Sistem de monitorizare a calitatii aerului

Studenti: Razaila Geanina-Alexandra

Arazan Georgiana

Timisoara

Anul 2020

Continut

Introducere	4
Analiza domeniului	5
Tehnologii folosite	6
Java	6
Baza de date	6
Versionarea codului	6
Android Studio	7
Componente hardware folosite	7
DHT11	7
MQ-2	8
Calibrarea senzorului de gaz	9
Senzor Sharp	10
Placuta Arduino	11
Modul Bluetooth BT06	12
Proiectarea sistemului	13
Diagrama UML a cazurilor de utilizare	13
Actorii	14
Cazurile de utilizare pentru actori	14
Proiectarea bazei de date	15
Diagrama UML a bazei de date	15
Securitatea bazei de date	16
Implementarea functionalitatilor	18
Inregistrarea in aplicatie	18
Autentificarea in aplicatie	23
Programarea sistemului hardware	28
Utilizarea aplicatiei	30
Inregistrarea si autentificarea in aplicatie	30
Alegerea afectiunilor si setarea parametrilor	32
Vizualizarea profilului si a istoricului	33
Vizualizarea valorilor masurate	34

Pagina de administrator	36
Concluzii.....	37
Bibliografie	38

Introducere

Trăim într-o lume din ce în ce mai agitată pentru care calitatea produselor folosite e importantă. Dacă punem atât de multă importanță pe calitatea produselor folosite, cu atât mai mult trebuie să fim atenți la ceea ce ne înconjoară și la ce folosim în fiecare secundă a vieții noastre, și anume aerul pe care îl respirăm. Fie că avem o alergie, suferim de astm sau doar vrem să vedem anumite informații despre aerul pe care îl respirăm, avem nevoie de un sistem să măsoare acești parametrii pentru noi.

La o simplă căutare pe internet putem să vedem că una din cinci persoane suferă de o alergie sau că în Europa peste 30 de milioane de oameni suferă de astm. Am încercat să ne punem puțin în locul celor care au o astfel de problemă și ne-am gândit că un lucru care ne-ar ajuta este un sistem de monitorizare a aerului pentru a putea afla ce anume din aer ne deranjează și să îl eliminăm.

Proiectul pe care noi l-am dezvoltat măsoară temperatura, umiditatea, nivelul de fum, de gaz(GPL) și de praf din aer și le transmite prin intermediul bluetooth-ului aplicației Android pentru a putea fi vizualizate într-un mod cât mai prietenos utilizatorului. Dacă valorile pentru unul dintre parametrii măsurați depășesc limitele normale, utilizatorul o să fie instintat de acest lucru pentru a se putea proteja.

Analiza domeniului

La o simpla cautare in Google Play, gasim o multime de aplicatii care masoara calitatea aerului. Pentru a putea scoate in evidenta avantajele aplicatiei noastre, o sa facem o comparatie intre aceasta si alte doua aplicatii asemanatoare gasite pe internet: Airveda-Air Quality & Asthma Tracker cu peste 50000 de descarcari si Air Quality|AirVisual cu peste 5 milioane de descarcari.

Caracteristici si functionalitati	Airveda-Air Quality & Asthma Tracker	Air Quality AirVisual	Air monitorizer
Sistem de operare	Android	Android	Android
Magazin de aplicatii	Google Play	Google Play	x
Nota din Google Play	3.6/5	4.7/5	x
Numar de instalari	50K	5M	x
Creare cont	Da	Nu	Da
Parametrii masurati	Temperatura, Umiditate, PM2.5, PM10, Indicele de calitate a aerului	Temperatura, Umiditate, PM2.5, PM10, Indicele de calitate a aerului , NO2	Temperatura, Umiditate, Gaz, Praf, Fum
Istoric	Da	Da	Da
Monitorizarea aerului din camera	Nu	Nu	Da

Sistemul nostru hardware este formata dintr-un senzor DHT11 care ne permite sa masuram temperatura si umiditatea, dintr-un senzor MQ-2 care ne permite sa masuram nivelul de fum si de gaz din camera si dintr-un senzor de particule care determina praful din aer. Datorita faptului ca aplicatia noastra comunica direct, prin intermediul bluetooth-ului, cu instalatia hardware, noi putem sa amplasam aceasta instalatie oriunde dorim sa aflam informatii despre starea aerului ceea ce reprezinta principalul avantaj fata de alte aplicatii care preiau datele de la anumite statii de monitorizare aflate in anumite orase(daca nu ai statie de monitorizare la tine in oras, o sa preia de la cea mai apropiata statie). Crearea unui cont permite fiecarui utilizator sa aiba o evidenta a caracteristicilor aerului masurate in anumite dati.

Tehnologii folosite

Java

Pentru crearea aplicatiei am ales sa scriem codul in Java folosind Android Studio.

Java este un limbaj de programare orientat pe obiecte, puternic tipizat, conceput de către James Gosling la Sun Microsystems (acum filială Oracle) la începutul anilor '90, fiind lansat în 1995. Limbajul împrumută o mare parte din sintaxă de la C și C++, dar are un model al obiectelor mai simplu și prezintă mai puține facilități de nivel jos. Un program Java compilat, corect scris, poate fi rulat fără modificări pe orice platformă care e instalată o mașină virtuală Java.

Baza de date

Ca baza de date am ales sa folosim o baza de date in timp real folosind Firebase pentru ca sincronizarea datelor între utilizatori în timp real este ușoară, oferă flexibilitate pentru a accesa date de pe orice dispozitiv si un alt avantaj este că poate fi utilizat și offline. Când conexiunea este pierdută, baza de date utilizează memoria cache locală pe dispozitiv pentru a stoca modificările.

Firebase este o platformă de dezvoltare a aplicațiilor mobile și web dezvoltată de Firebase Inc. în 2011, achiziționată apoi de Google în 2014. Începând cu octombrie 2018, platforma Firebase are 18 produse care sunt folosite de peste 1,5 milioane de aplicații.

Stochează datele în format JSON (JavaScript Object Notation), care nu utilizează interogarea pentru inserarea, actualizarea, ștergerea sau adăugarea de date. Este backend-ul unui sistem care este folosit ca bază de date pentru stocarea datelor.

Versionarea codului

Pentru versionarea codului am ales sa folosim Git si GitHub.

GitHub, ca și definiție, este un sistem de versionare (version-control) care urmărește modificările fișierelor și care poate ajuta colaborarea între mai multe persoane care lucrează la un anumit proiect.

GitHub-ul este o platformă care ne oferă Git-ul ca și serviciu web. Astfel, noi putem să ne salvăm repo-urile online, și să avem un link de acces la ele oricând avem nevoie.

Link-ul catre proiectul nostru este urmatorul:

<https://github.com/GeorgiANA81/AirMonitorizer.git>

Android Studio

Aplicatia noastra este construita in Android Studio. Android Studio este un mediu de dezvoltare pentru colaborarea cu platforma Android. Android Studio este bazat pe software-ul IntelliJ IDEA de la JetBrains, este instrumentul oficial de dezvoltare a aplicațiilor Android. Acest mediu de dezvoltare este disponibil pentru Windows, OS X și Linux.

Componente hardware folosite

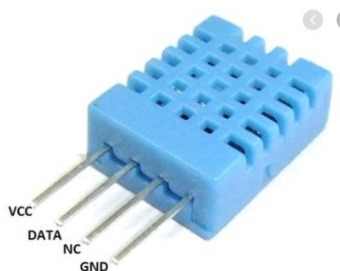
DHT11

DHT11 este un senzor digital de temperatură și umiditate, care are incorporat un senzor de umiditate capacitiv și un termistor, pentru a măsura aerul din jur și dă un semnal digital pe pinul de date (nu necesită pini de intrare analogici LCD).

Senzorul DHT11 oferă o calibrare extrem de precisă a camerei de calibrare a umidității. Coeficienții de calibrare sunt stocați în memoria programului OTP, senzorii interni detectează semnalele în proces. Sistemul de interfață serială cu un singur fir este integrat pentru a fi rapid și ușor. Are dimensiuni reduse, cu o putere redusă ce pot transmite semnalul de până la 20 de metri.

Caracteristici tehnice:

- Poate măsura umiditatea în intervalul 20% - 90% cu o precizie de 5%
- Poate măsura temperatura în intervalul 0 – 50° C cu o precizie de 2° C.
- Tensiune de alimentare: 3.3V - 5V;
- Curent: 2.5mA (maxim);
- Gama de măsurare a umidității: 20% - 95% RH;
- Acuratețea măsurării umidității: $\pm 5\%$ RH;
- Gama de măsurare a temperaturii: 0° C - 60° C;
- Acuratețea măsurării temperaturii: $\pm 2^{\circ}\text{C}$;
- Atenție! Nu funcționează sub 0° C.
- Dimensiuni: 32mm x 14mm.



MQ-2

Modulul este folosit pentru a detecta scurgerile de gaze în încăperi mici sau mari și reprezintă o metodă de precauție pentru incendii sau pentru intoxicații.

Senzorul are o sensibilitate ridicată și principalele gaze pe care le vizează sunt GPL-ul, izobutan, propan, metan, alcool, hidrogen și fum.

Senzorul dispune de un comparator, astfel că puteți citi date analogice în timp real sau puteți afla dacă concentrația de gaz a depășit o anumită limită.

Caracteristici tehnice:

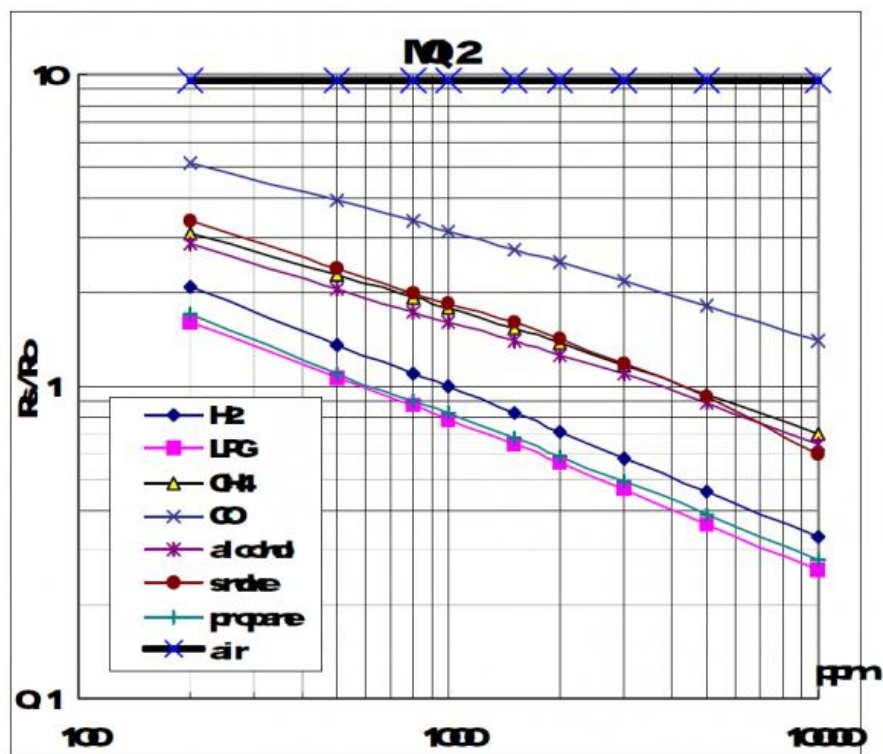
- Tensiune: 5V;
- Curent: 150mA;
- Rezistență heater: 33R;
- Temperatură de lucru: -20°C - 50°C;
- Output digital sau analogic.

Dimensiuni: 33 x 14 x 1.6mm;



Calibrarea senzorului de gaz

Concentrația unui gaz este dată în PPM (părți per milion). O dificultate a lui MQ-2 este aceea că se dă o singură valoare analogică cu care conținutul de gaz din aer trebuie să fie calculat pentru diferitele gaze suportate. Totuși, senzorul trebuie să fie configurat în acest scop. În primul rând, trebuie să vedem fișa tehnică a respectivului modul, care conține următoarea diagramă:



Dar, scalarea valorilor nu este liniară, ci logaritmică în baza 10 (log), astfel, prima linie pe axa X este la 200, apoi 300, etc. Prima linie după 1000 este 2000, etc. Distanța dintre ele este liniară. Ideea din spatele acestui script pentru calibrare și citire este de a crea o linie dreaptă și a calcula cantitatea de gaz (în ppm). Pentru a face acest lucru, avem nevoie de două puncte pentru a calcula panta.

Să luăm exemplul GPL. Prin urmare, luăm punctul P1 ($x = 200$, $y = \sim 1,62$) și P2 ($x = 10000$, $y = \sim 0,26$). Pentru a calcula valorile "reale", aplicăm logaritm zece. Folosind formula cu două puncte, putem calcula panta, care în cazul nostru este -0.47 (legătura la calcul). Cu panta și logaritmul calculat din punctul stâng ($x = 2,3$, $y = 0,21$), putem determina acum linia dreaptă.

Panta m a unei drepte pe care sunt fixate punctele $A(x_1, y_1)$ și $B(x_2, y_2)$ este:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

Pentru gazele rămase, calculul este echivalent și poate fi efectuat în același mod.

Senzor Sharp

Acest senzor optic poate detecta particule foarte fine cum ar fi fumul de tigara, de asemenea este efectiv si des folosit in sistemele de purificare a aer.

Acest senzor are un consum de mic de curent(pana la 20mA, 11mA in general) si poate duce un senzor de pana la 7V DC. Iesirea este analogica cu un voltaj proportional cu concentratia de praf masurata si cu sensibilitatea de 0.5V/0.1mg/m³.

SPECIFICATII

Temperatura de lucru: -10 la 60 grade Celsius

Temperatura de depozitare: -20 la 60 grade Celsius

Detecteaza particule mai mari de 0.8 microni

Consum total de curent: 20 mA maxim, 11mA de obicei

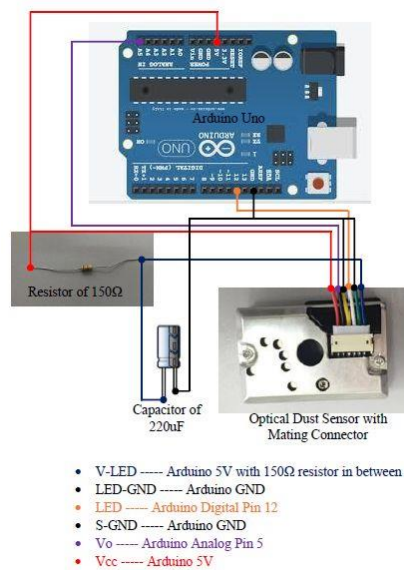
Iesirea este analogica cu un voltaj proportional cu concentratia de praf masurata si cu sensibilitatea de 0.5V/0.1mg/m³.

Tensiunea necesara pentru operare: 5-7 V

Durata de viata: 5 ani

Dimensiuni: 46mm * 30mm * 17.6mm

Greutate: 15g



Placuta Arduino

Arduino UNO constituie o platforma de procesare tip open-source, bazata pe un software si hardware flexibil construita in jurul unui microcontroler ATMEGA 328P-PU capabila de a prelua date printr-o serie de senzori conectati la pinii placii si de a actiuna asupra altor dispozitive ca LED-uri, motoarelor, servomotoare, sau alte tipuri de dispozitive mecanice pe baza unor comenzi cuprinse in codul scris intr-un limbaj de programare, similar cu limbajul C++ incarcata in memoria microcontrolerului.

Microcontrolerul de pe placa Arduino poate fi programat prin mediul Arduino, daca se selecteaza Arduino Uno din menu-ul Tools, alegand corect tipul microcontrolerului de pe placa. De remarcat este ca microcontrolerul ATmega328 de pe Arduino Uno, vine incarcata cu un bootloader care ofera posibilitatea ca sa se poata incarca noul cod, fara utilizarea unui dispozitiv hardware suplimentar. Comunicatia are loc folosind protocolul STK500. Daca se doreste programarea fara folosirea bootloader-ului, se poate folosi pinii ICSP (In-Circuit Serial Programming). Aceasta metoda ofera avantajul de a castiga 0,5 kB din memorie, care in mod normal este ocupata de bootloader, insa are dezavantajul de a avea nevoie de un programator extern.



Modul Bluetooth BT06

Modul Bluetooth BT06, compatibil Arduino si alte placi de dezvoltare.

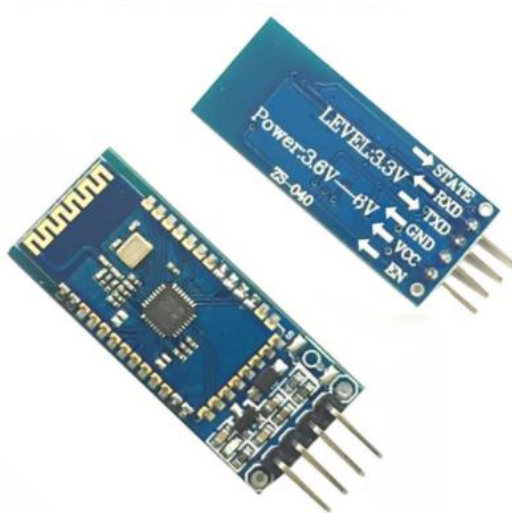
Modulul suporta interfata UART si protocolul serial port SPP. Avantajele lui sunt dimensiunile reduse, costul si inalta sensibilitate al trans-receiverului.

Specificatii:

Bluetooth V3.0 + EDR

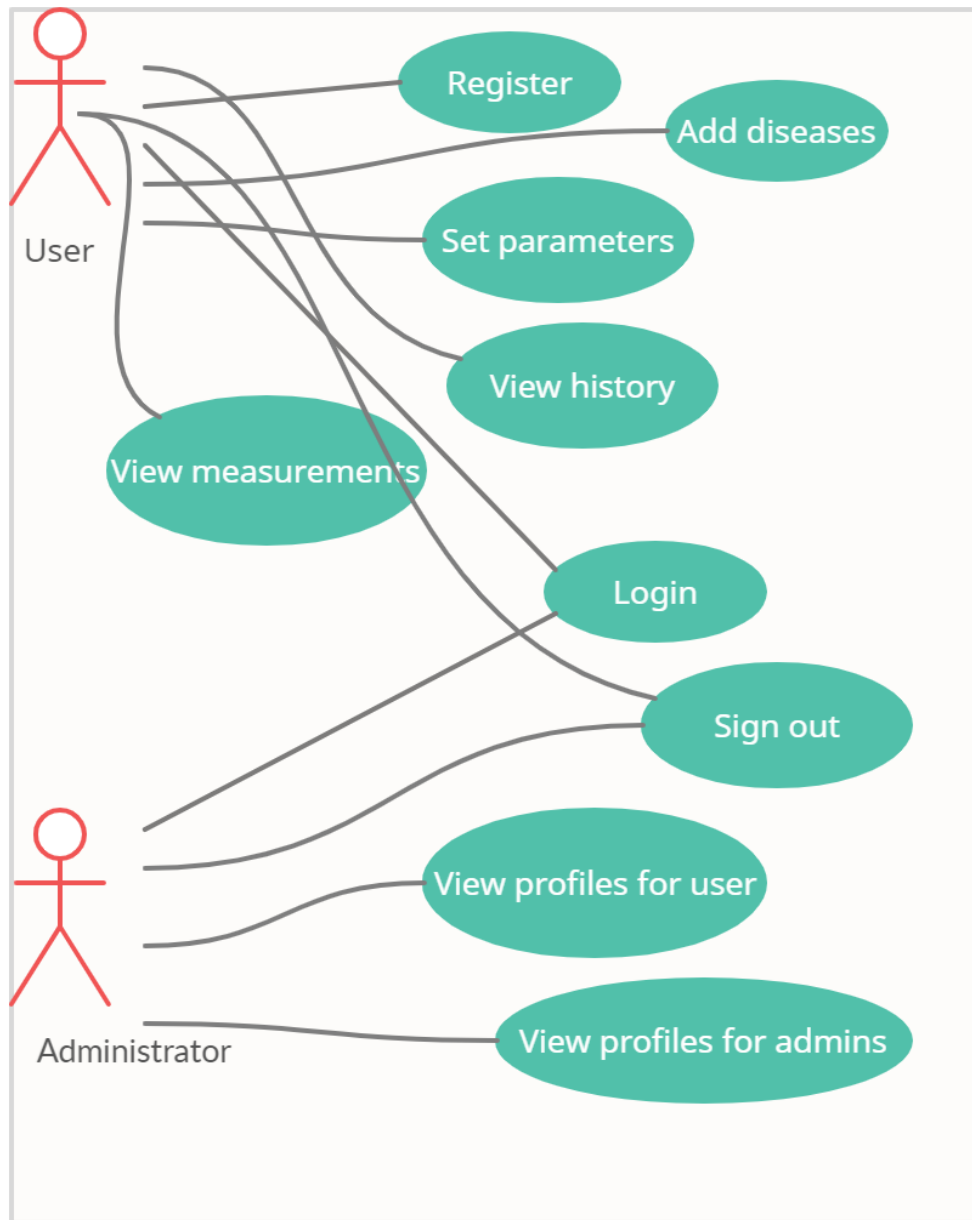
Bluetooth Class 2

Tensiune alimentare: 3.3V



Proiectarea sistemului

Diagrama UML a cazurilor de utilizare



Actorii

Acest sistem are 2 actori: utilizatorii aplicatiei si administratorul.

Administratorii sunt autentificati pe baza unei caracteristici salvate in baza de date si o sa aiba o interfata a aplicatiei diferita fata de utilizatorii normali pentru a putea face schimbari asupra aplicatiei.

Utilizatorii aplicatiei o sa poata folosi aplicatia doar dupa ce si-au creat un cont, urmand ca dupa acest pas sa poate sa vizualizeze diferite informatii, dar nu sa faca modificari in aplicatie.

Cazurile de utilizare pentru actori

Fiecare actor poate sa efectueze anumite operatii in aplicatie dintre care unele sunt comune, iar altele total diferite.

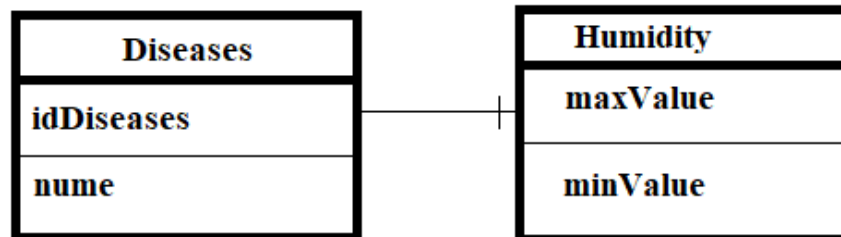
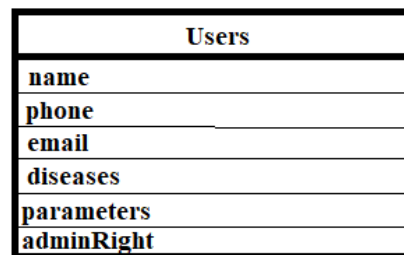
Cazurile de utilizare comune atat pentru utilizatori, cat si pentru administratori, sunt autentificarea in aplicatie si iesirea din aplicatie(delogarea).

Administratorii pot sa vizualizeze profilurile tuturor utilizatorilor care si-au creat un cont, si de asemenea, sa vada cine mai este administrator

Ca utilizator, cazurile de utilizare sunt:crearea unui cont, setarea bolilor de care sufera, alegerea parametrilor care sa fie masurati, vizualizarea datelor preluate de la senzori, vizualizarea unui istoric a datelor masurate.

Proiectarea bazei de date

Diagrama UML a bazei de date



Securitatea bazei de date

Firebase este certificat conform standardelor majore de confidențialitate și securitate.

Toate serviciile Firebase (în afară de Distribuția aplicațiilor și Crashlytics) au finalizat cu succes procesul de evaluare ISO 27001 și SOC 1, SOC 2 și SOC 3, iar unele au finalizat și procesul de certificare ISO 27017 și ISO 27018:

Service name ▼	ISO 27001	ISO 27017	ISO 27018	SOC 1	SOC 2	SOC 3
Cloud Firestore	✓	✓	✓	✓	✓	✓
Cloud Functions for Firebase	✓	✓	✓	✓	✓	✓
Cloud Storage for Firebase	✓	✓	✓	✓	✓	✓
Firebase A/B Testing	✓			✓	✓	✓
Firebase App Distribution					✓	
Firebase Authentication	✓	✓	✓	✓	✓	✓
Firebase Cloud Messaging	✓			✓	✓	✓
Firebase Crashlytics					✓	
Firebase Dynamic Links	✓			✓	✓	✓
Firebase Hosting	✓			✓	✓	✓
Firebase In-App Messaging	✓			✓	✓	✓
Firebase ML	✓			✓	✓	✓
Firebase Performance Monitoring	✓			✓	✓	✓

Serviciile Firebase criptează datele în tranzit folosind HTTPS și izolează logic datele clienților.

Pentru a păstra datele personale în siguranță, Firebase folosește măsuri extinse de securitate pentru a minimiza accesul:

->Firebase restricționează accesul la angajații selectați care au un scop comercial de a accesa datele personale.

->Firebase înregistrează accesul angajaților la sistemele care conțin date cu caracter personal.

->Firebase permite accesul la datele personale numai de către angajații care se conectează cu autentificare Google și autentificare cu 2 factori(email+parola).

Atunci cand un nou utilizator isi creaza un cont, adresa lui de email o sa se salveze in Firebase Authentication, insa parola o sa fie criptata si nu o sa apara in baza de date, astfel incat nici macar administratorul bazei de date nu poate sa o vada(nici macar criptata):

Search by email address, phone number, or user UID					Add user	↺	⋮
Identifier	Providers	Created	Signed In	User UID ↑			
razaila.alexandra@yahoo.com	✉	Nov 27, 2020	Nov 27, 2020	JLUukK634CRxayK6hUhYrqH0qbk2			
ion.popescu@yahoo.com	✉	Nov 27, 2020	Nov 27, 2020	Yxdf7GMB2gWuTnpVYv9I4TYrIOf2			
ana.popa@yahoo.com	✉	Nov 27, 2020	Nov 27, 2020	e1qP15Y8ufWZTnp96nB8ylE6ZCY2			
arazan_georgiana@yahoo.com	✉	Nov 28, 2020	Nov 28, 2020	ks4gycXMF1MXdaGNh8Gw67c9f...			
Rows per page: 50					1 - 4 of 4	⏪	⏩

Restul informatiilor pentru utilizatori o sa fie salvate in baza de date in timp real, cu exceptia parolei:

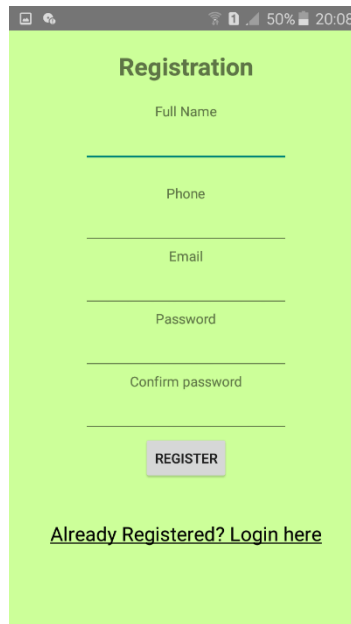


Implementarea functionalitatilor

Inregistrarea in aplicatie

Primul pas in folosirea acestei aplicatii o constituie crearea unui cont.

Pentru crearea unui cont, utilizatorul trebuie sa completeze un formular de inregistrare care contine urmatoarele campuri: numele intreg(nume+prenume), numarul de telefon, adresa de email si parola(plus un camp pentru confirmarea parolei), dupa cum se poate vedea si in figura urmatoare:

A screenshot of a mobile application's registration screen. The screen has a light green background. At the top, there's a status bar with icons for signal, battery (50%), and time (20:08). Below the status bar, the word "Registration" is centered in a bold, dark green font. Underneath, there are five input fields, each with a label above it: "Full Name", "Phone", "Email", "Password", and "Confirm password". Each input field is represented by a horizontal line. Below these fields is a purple button with the word "REGISTER" in white capital letters. At the bottom of the form, there is a link that says "Already Registered? Login here" in a dark green font.

In cazul in care utilizatorul nu completeaza toate informatiile cerute si apasa butonul de inregistrare, o sa fie atentionat ca nu a completat toate campurile si nu o sa poata sa isi creeze un nou cont. De asemenea, utilizatorul trebuie sa introduca o adresa de email valida(care sa fie sub forma: nume@yahoo/gmail.com) si o parola de minim 6 caractere. Pentru fiecare camp, in cazul introducerii unor informatii gresite sau a lipsei informatiei, o sa fie afisat un mesaj de atentionare. De asemenea, daca parola din campul "Confirm password" este diferita de cea de la campul "Password", o sa fie avertizat ca parolele nu coincid.

The screenshot shows a registration form with a light blue background. The title is "Registration". The first field is "Full Name", which is empty. A red error message "Full name is required!" is displayed below the field. The other fields are "Email", "Password", and "Confirm password", all of which are empty. A "REGISTER" button is at the bottom, and a link "Already Registered? Login here" is below it.

The screenshot shows the same registration form, but now with data entered. The "Full Name" field contains "Razaila Alexandra". The "Phone" field contains "0766676666". The "Email" field contains "razaila", and a red error message "Please provide valid email!" is displayed below it. The "Password" and "Confirm password" fields are still empty. The "REGISTER" button and the "Already Registered? Login here" link are still present.

Datele pe care le completam sunt salvate in asa numitele blocuri EditText de la care o sa preluam informatia in fisierul “activity” si o sa eliminam posibilele spatii goale de la inceput si de la final(pentru acest lucru am folosit functia trim()) si apoi le verificam sa fie corect completate.

```
private void registerUser() {

    String name = editText1.getText().toString().trim();

    String phone = editText2.getText().toString().trim();

    String email = editText3.getText().toString().trim();

    String password1 = editText4.getText().toString().trim();

    String password2 = editText5.getText().toString().trim();

    if(name.isEmpty()){

        editText1.setError("Full name is required!");

        editText1.requestFocus();

        return;

    }

    if(phone.isEmpty()){

        editText2.setError("Phone number is required!");

        editText2.requestFocus();

        return;

    }

    if(email.isEmpty()){
```

```

        editText3.setError("Email is required!");

        editText3.requestFocus();

        return;
    }

    if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){

        editText3.setError("Please provide valid email!");

        editText3.requestFocus();

        return;
    }

    if(password1.isEmpty()){

        editText4.setError("Password is required!");

        editText4.requestFocus();

        return;
    }

    if(password1.length() < 6){

        editText4.setError("Minimum password length should be 6 characters!");

        editText4.requestFocus();

        return;
    }

    if(password2.isEmpty()){

        editText5.setError("Confirm password is required!");

        editText5.requestFocus();

        return;
    }

    if(!password1.equals(password2)){

        editText5.setError("Invalid password!");

        editText5.requestFocus();

        return;
    }
}

```

In momentul in care toate campurile sunt completate corect si este apasat butonul de inregistrare, datele utilizatorului o sa se salveze in baza de date Firebase.

Prima data se salveaza email-ul si parola in Firebase Authentication cu ajutorul functiei `createUserWithEmailAndPassword(email, password1)`, urmand ca apoi sa se salveze celelalte informatii despre utilizator in baza de date in timp real. Informatiile care se salveaza in baza de date in timp real sunt salvate initial intr-un obiect de tipul clasei `User`. In baza de date o sa fie o cheie “Users” care o sa contina mai multe coduri care fiecare reprezinta un utilizator, avand salvat numele, numarul de telefon si email-ul. Dupa ce noul cont s-a creat cu succes, utilizatorul o sa primeasca un mesaj prin care este felicitat pentru crearea noului cont. Daca apare o eroare, utilizatorul o sa fie avertizat si rugat sa incerce din nou.

Deoarece pot sa existe persoane cu acelasi nume, inregistrarea o sa se faca in functie de email(care este unic), astfel daca doi utilizatori cu acelasi nume vor sa foloseasca acelasi model de email, cel care o sa incerce sa isi creeze contul ultimul o sa fie atentionat ca exista deja o inregistrare cu acea adresa de email.

Pentru a verifica existenta unei adrese de email, folosim blocul try-catch pentru a trata exceptia cand sunt doua adrese la fel(*FirebaseAuthUserCollisionException*).

[illegible]

```

        Toast.makeText(RegistrationActivity.this, "Failed to register! Try again!",
Toast.LENGTH_LONG).show();

    }

}

});

}

else{

    progress.setVisibility(View.GONE);

    try

    {

        throw task.getException();

    }

    catch(FirebaseAuthUserCollisionException existEmail)

    {

        Toast.makeText(RegistrationActivity.this, "This email is already used!Please use another
email!", Toast.LENGTH_LONG).show();

    }

    catch(Exception e)

    {

        Toast.makeText(RegistrationActivity.this, "Failed to register!Try again later!",
Toast.LENGTH_LONG).show();

    }

}

}

}

});

```

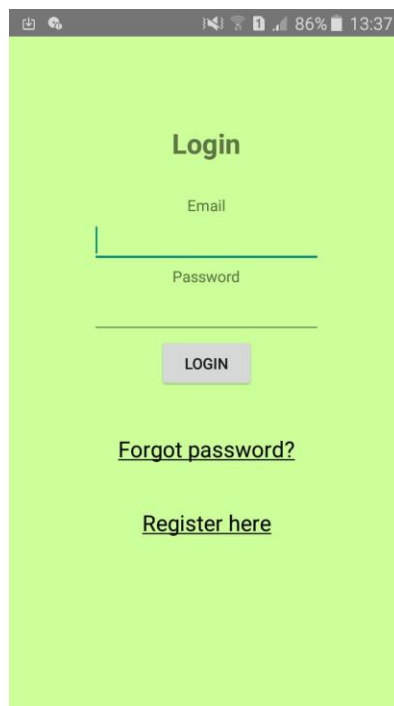
Autentificarea in aplicatie

Autentificarea in aplicatie se realizeaza cu ajutorul adresei de email si a parolei(dupa cum putem vedea in figura de mai jos). Daca utilizatorul nu completeaza unul din cele doua campuri, o sa fie atentionat prin intermediul unor mesaje.

In momentul autentificarii, campul “admin” din baza de date este verificat si in functie de valoarea sa, utilizatorul o sa fie redirectionat catre pagina de administrator(daca admin=”yes”) sau catre pagina de utilizator(daca admin=”no”). Pentru a verifica acest camp, salvam id-ul utilizatorului curent din Firebase Authentication, il cautam in baza de date in timp real si preluam valoarea pentru campul “admin”.

Email-ul si parola le preluam din doua blocuri EditText, eliminam posibilele spatii goale de la inceput si sfarsit si le verificam daca sunt completate corect sau nu. Daca campurile sunt completate corect, o sa fim redirectionati catre pagina de start unde trebuie sa ne alegem afectiunile de care suferim, parametrii pe care vrem sa ii masuram, etc.

Pentru a verifica daca adresa introdusa exista in baza de date si parola e corecta folosim un bloc try-catch prin care atentionam utilizatorul in cazul in care apar anumite exceptii:parola gresita(*FirebaseAuthInvalidCredentialsException*) sau email invalid(*FirebaseAuthInvalidUserException*).



```

private void userLogin() {

    String email = editText1.getText().toString().trim();

    String password = editText2.getText().toString().trim();


    if(email.isEmpty()){

        editText1.setError("Email is required!");

        editText1.requestFocus();

        return;

    }

    if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){

        editText1.setError("Please provide valid email!");

        editText1.requestFocus();

        return;

    }

    if(password.isEmpty()){

        editText2.setError("Password is required!");

        editText2.requestFocus();

        return;

    }

    if(password.length() < 6){

        editText2.setError("Minimum password length should be 6 characters!");

        editText2.requestFocus();

        return;

    }

    mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {

        @Override

        public void onComplete(@NonNull Task<AuthResult> task) {

            progress.setVisibility(View.VISIBLE);

            if(task.isSuccessful){

                //redirect to user profile

                user = FirebaseAuth.getInstance().getCurrentUser();

```



```

rootRef = FirebaseDatabase.getInstance().getReference("Users");
userID = user.getId();
rootRef.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        User userProfile = snapshot.getValue(User.class);
        if(userProfile!=null){
            admin = userProfile.admin;
        }

        if(admin.equals("yes")){
            //admin profile
            progress.setVisibility(View.GONE);
            startActivity(new Intent(LoginActivity.this, adminPageActivity.class));
        }
        else{
            //user profile
            progress.setVisibility(View.GONE);
            startActivity(new Intent(LoginActivity.this, WelcomeUserActivity.class));
        }
    }
});

@Override
public void onCancelled(@NonNull DatabaseError error) {
    progress.setVisibility(View.GONE);
    Toast.makeText(LoginActivity.this, "Something went wrong!",
Toast.LENGTH_LONG).show();
}

}

else {
    progress.setVisibility(View.GONE);

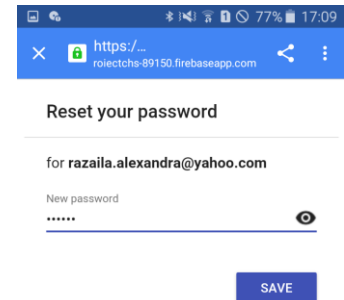
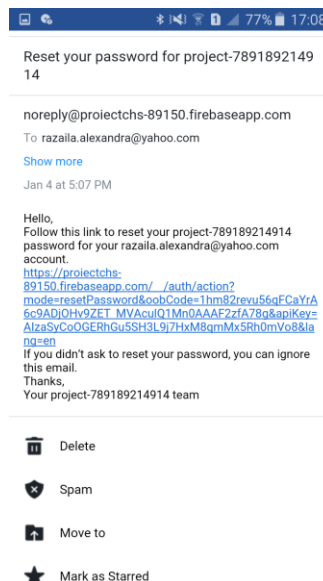
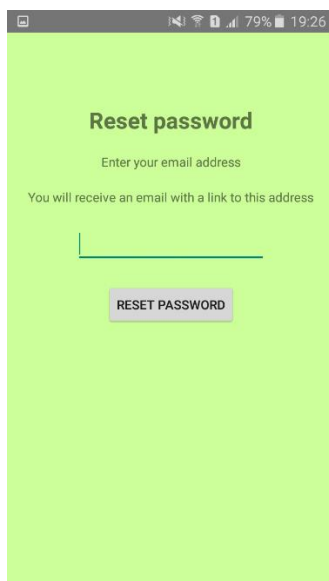
```

```

    try {
        throw task.getException();
    } catch (FirebaseAuthInvalidUserException invalidEmail) {
        Toast.makeText(LoginActivity.this, "You entered a wrong email!",
Toast.LENGTH_LONG).show();
    } catch (FirebaseAuthInvalidCredentialsException invalidPassword) {
        Toast.makeText(LoginActivity.this, "You entered a wrong password!",
Toast.LENGTH_LONG).show();
    } catch (Exception e) {
        Toast.makeText(LoginActivity.this, "Failed to login! Please try again!",
Toast.LENGTH_LONG).show();
    }
}
}
});
}

```

Daca utilizatorul isi uita parola, are posibilitatea sa isi reseteze parola prin intermediul adresei de email unde o sa primeasca un link catre o pagina in care sa isi introduca noua parola. La fel ca in cazurile demai sus, adresa de email o preluam dintr-un bloc de tip EditText, eliminam spatiile goale(daca exista la inceput si sfarsit), verificam daca a copmpletat corect blocul si apoi apelam functia `sendPasswordResetEmail(emailText)` pentru a trimite link-ul cu resetarea parolei la adresa respectiva.



```

private void resetPassword() {
    String emailText = email.getText().toString().trim();
    if(emailText.isEmpty()){
        email.setError("Email is required!");
        email.requestFocus();
        return;
    }
    if(!Patterns.EMAIL_ADDRESS.matcher(emailText).matches()){
        email.setError("Please provide valid email!");
        email.requestFocus();
        return;
    }
    mAuth.sendPasswordResetEmail(emailText).addOnCompleteListener(new OnCompleteListener<Void>()
{
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            Toast.makeText(ResetPasswordActivity.this, "Check your email!", Toast.LENGTH_LONG).show();
            startActivity(new Intent(ResetPasswordActivity.this, LoginActivity.class));
        }
        else{
            Toast.makeText(ResetPasswordActivity.this, "Try again! Something wrong happened!",
Toast.LENGTH_LONG).show();
        }
    }
});
}
}

```

Programarea sistemului hardware

Pentru a putea prelua datele de la senzori, trebuie sa programam placuta Arduino astfel incat sa faca ce vrem noi. Am conectat senzorul dht11 la pinul digital 2 de pe placa, senzorul MQ2 la pinul analog A0, iar senzorul de praf e conectat la pinul analog A1 si pinul digital 4.

```
#define DHTPIN 2
#define DHTTYPE DHT11
#define MQ_PIN (0)

int measurePin = 1; //senzorul de praf e conectat la pinul A1 al placii Arduino
int ledPower = 4; //led-urile senzorului de praf sunt conectate la pinul D4
```

Pentru a citi temperatura si umiditatea de la senzorul dht11 folosim 2 functii predefinite din biblioteca "DHT.h":

```
float h = dht.readHumidity(); // citim umiditatea
float t = dht.readTemperature(); // citim temperatura in grade Celsius
```

Pentru a calcula cantitatea de praf din aer se folosesc urmatoarele formule pentru senzorul de praf:

```
voMeasured = analogRead(pinulAnalogDeLaSenzorulDePraf);
calcVoltage = voMeasured*(5.0/1024);
dustDensity = 0.17*calcVoltage-0.1;
```

Pentru a determina cantitatea de fum si de gaz din aer preluate de la senzorul MQ2, folosim cateva functii create de noi:

```
float MQRead(int mq_pin)
{
    int i;
    float rs=0;
    for (i=0;i<READ_SAMPLE_TIMES;i++) {
        rs += MQResistanceCalculation(analogRead(mq_pin));
        delay(READ_SAMPLE_INTERVAL);
    }
}
```

```

}

rs = rs/READ_SAMPLE_TIMES;

return rs;

}

int MQGetGasPercentage(float rs_ro_ratio, int gas_id)
{
    if( gas_id == GAS_LPG ) {
        return MQGetPercentage(rs_ro_ratio,LPGCurve);
    } else if( gas_id == GAS_CO ) {
        return MQGetPercentage(rs_ro_ratio,COCurve);
    } else if( gas_id == GAS_SMOKE ) {
        return MQGetPercentage(rs_ro_ratio,SmokeCurve);
    }

    return 0;
}

```

```

int MQGetPercentage(float rs_ro_ratio, float *pcurve)
{
    return (pow(10,((log(rs_ro_ratio)-pcurve[1])/pcurve[2]) + pcurve[0]]));
}

```

Pentru a transmite datele prin bluetooth, verificam daca conexiunea este realizata si apoi transmitem un mesaj cu toate informatiile preluate de la senzori, urmand ca in aplicatie sa fie separate:

```

if(Serial.available()){
    Serial.println(String(t)+","+String(h)+","+
        +MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_LPG)+","+
        +dustDensity+",",
        +MQGetGasPercentage(MQRead(MQ_PIN)/Ro,GAS_SMOKE)+",");
}

```

Utilizarea aplicatiei

Inregistrarea si autentificarea in aplicatie

Prima pagina pe care o vede utilizatorul in momentul deschiderii aplicatiei o sa fie o pagina care are 2 butoane, unul care permite utilizatorului sa mearga catre pagina de autentificare si unul catre pagina de inregistrare (Figura1).

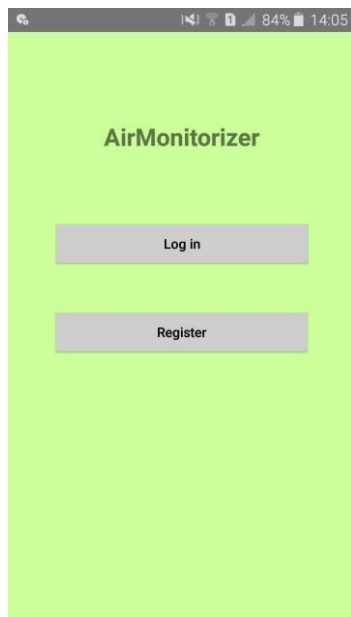


Figura 1:Prima pagina a
Aplicatiei

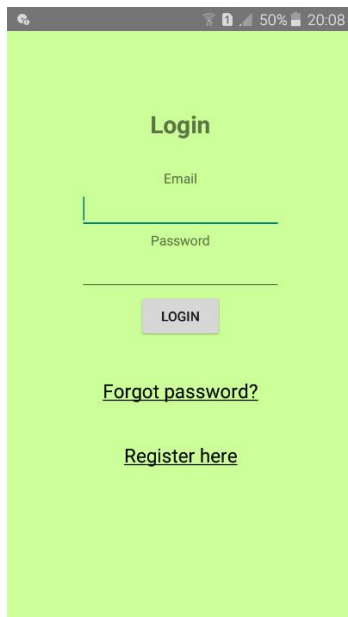


Figura 2: Pagina de
autentificare

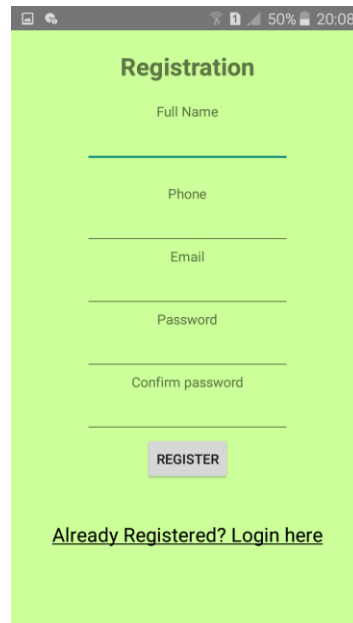


Figura 3:Pagina de
inregistrare

Daca utilizatorul nu are creat un cont, acesta trebuie sa isi creeze unul prin completarea unui formular care contine numele, numarul de telefon, adresa de email si parola(formularul din Figura 3). Pentru a putea utiliza aplicati1, orice utilizator este obligat sa isi creeze un cont, altfel nu va putea sa efectueze nicio alta activitate.

Daca un camp nu este completat sau nu corespunde cerintelor, utilizatorul o sa fie avertizat prin intermediul unor mesaje(Figura 4 si Figura 5) si nu o sa poata sa isi creeze un cont pana nu introduce datele corecte. De asemenea parola trebuie sa fie de minim 6 caractere pentru a fi considerata valida.

Registration

Full Name

Full name is required!

Email

Password

Confirm password

REGISTER

[Already Registered? Login here](#)

Figura 4: Camp necompletat

Registration

Full Name

Razaila Alexandra

Phone

0766676666

Email

razaila

Please provide valid email!

Confirm password

REGISTER

[Already Registered? Login here](#)

Figura 5: Adresa de email
invalida

Login

Email

ana

Please provide valid email!

LOGIN

[Forgot password?](#)

[Register here](#)

Figura 6: Adresa de email
invalida la autentificare

Daca utilizatorul are un cont creat anterior, el trebuie doar sa se autentifice in aplicatie prin intermediul adresei de email si a parolei(formularul din Figura 2). Daca nu introduce datele corecte, o sa fie avertizat ca si in cazul inregistrarii prin intermediul unor mesaje(Figura 6), iar daca datele introduse nu corespund unui cont existent in baza de date, o sa fie atentionat ca autentificarea a esuat(Figura 7).

Login

Email

razaila.alexandra@yahoo.com

Password

.....

LOGIN

[Forgot password?](#)

[Register here](#)

You entered a wrong email!

Figura 7: Autentificare esuata

Login

Email

razaila.alexandra@yahoo.com

Password

.....

LOGIN

[Forgot password?](#)

[Register here](#)

You entered a wrong password!

Figura 8: Resetarea parolei

Reset password

Enter your email address

You will receive an email with a link to this address

RESET PASSWORD

Email is required!

Reset password

Enter your email address

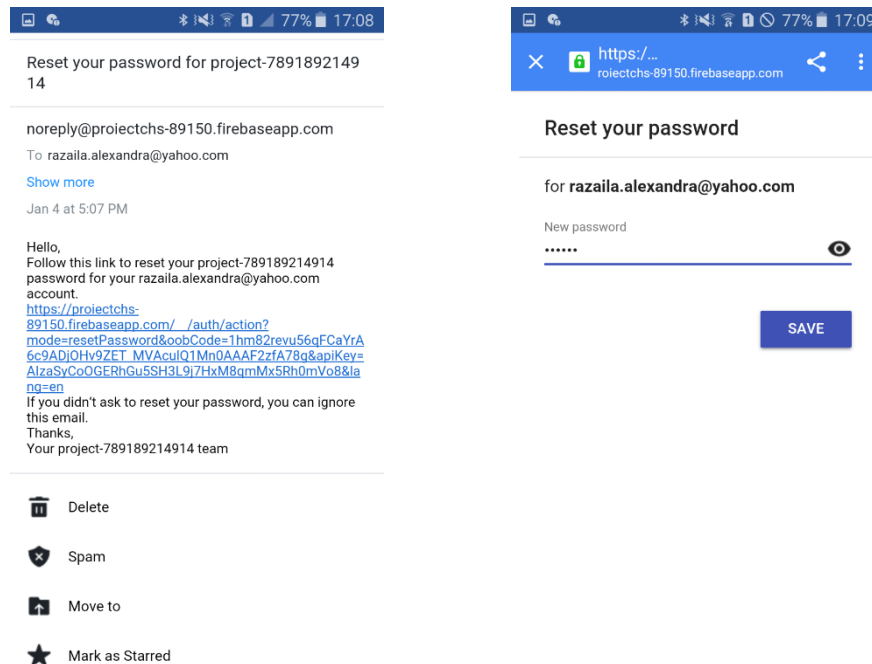
You will receive an email with a link to this address

RESET PASSWORD

Email is required!

Figura 9: Camp
necompletat

Formularul de autentificare din Figura 2 contine un buton care permite resetarea parolei in cazul in care utilizatorul nu isi mai aminteste parola. Prin apasarea acestui buton, utilizatorul o sa fie redirectionat catre formularul din figura 8 si o sa fie nevoit sa completeze formularul cu o adresa de email valida(in caz contrar primeste eroarea din figura 9) urmand ca apoi sa primeasca pe email un mesaj cu un link catre o pagina unde sa isi introduca noua parola.



Alegerea afectiunilor si setarea parametrilor

Dupe ce utilizatorul s-a autentificat, o sa fie redirectionat catre pagina unde o sa aiba posibilitatea sa isi aleaga afectiunile de care sufera: astm sau alergie la praf(Figura 10). Alegerea afectiunilor pot fi alese prin bifarea casutelor de langa denumirea afectiunilor, iar rezultatele o sa fie salvate in baza de date la apasarea butonului “Save”. Daca utilizatorul o sa se deconecteze si o sa se autentifice inapoi in aplicatie, alegerile o sa ramana salvate si o sa vada ce a bifat inainte sa iasa din aplicatie. Pe aceasta pagina se mai gaseste un buton care ne permite sa vedem detalii despre afectiunile prezente pe pagina(Figura 11). Pentru a putea sa vedem informatia completa, putem sa derulam in jos.

Daca apasam butonul “Set parameters” de pe coloana din stanga a ecranului, o sa ne apara o pagina de pe care putem sa ne alegem parametrii pe care vrem ca sistemul nostru sa ii masoare. Avem de ales dintre 5 parametrii: temperatura, umiditate, praf, gaz si fum. Alegerea lor se face la fel ca in cazul afectiunilor prin bifarea casutelor aferente fiecarei afectiuni si apasarea butonului “Save” pentru a fi salvate in baza de date. De asemenea, si aici o sa ramana alegerile salvate chiar daca ne deconectam si apoi intram din nou.

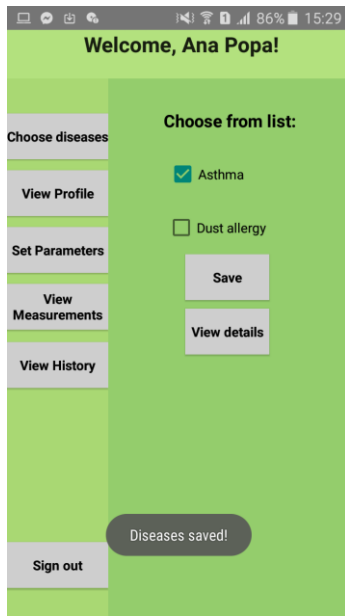


Figura 10: Alegerea
afectiunilor

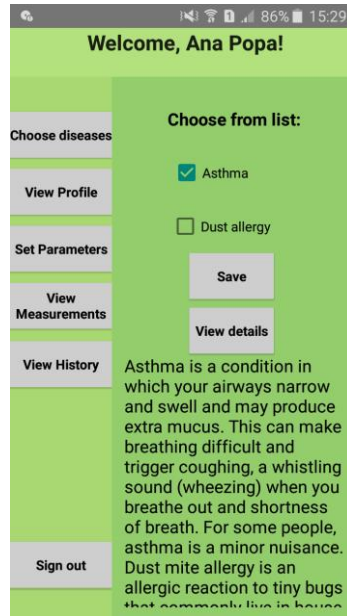


Figura 11: Vizualizarea
detaliilor despre afectiuni

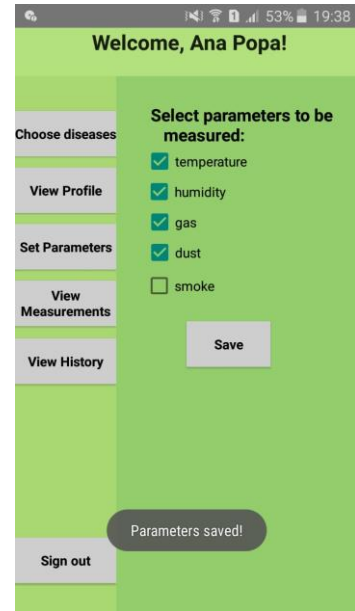


Figura 12: Setarea parametrilor

Vizualizarea profilului si a istoricului

Butonul “View profile” din partea stanga a ecranului ne permite sa vizualizam informatiile propriului profil: nume, numar de telefon, adresa de email, afectiunile alese si parametrii setati(Figura 14:b). Daca nu avem nimic selectat la parametrii sau la afectiuni, o sa ni se afiseze un mesaj informativ(Figura 13:a).

Butonul “History” din partea stanga a ecranului ne permite sa vedem istoricul masuratorilor in functie de ziua in care am intrat in aplicatie. Ca sa vedem toate masuratorile putem sa derulam in jos ca sa vedem textul integral(Figura 15). Butonul de Sign-out ne permite sa ne deconectam.

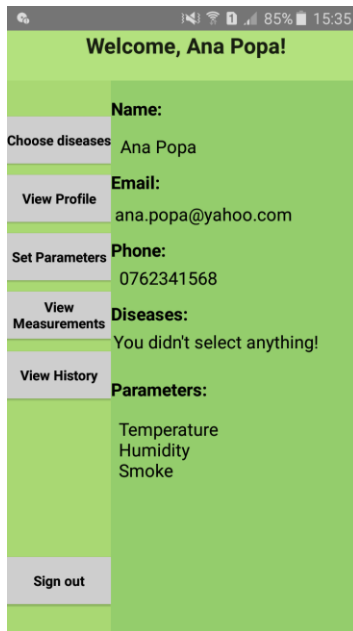


Figura 13: Vizualizare
profilului(a)

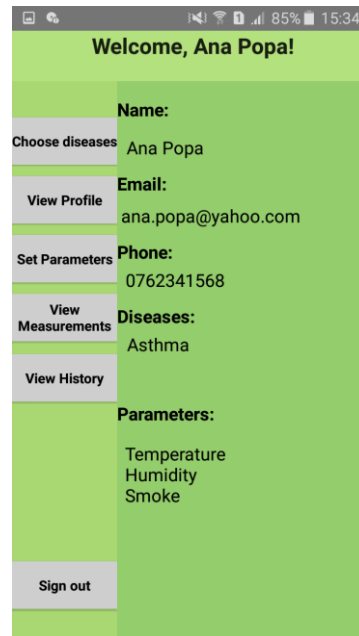


Figura 14: Vizualizarea
profilului(b)

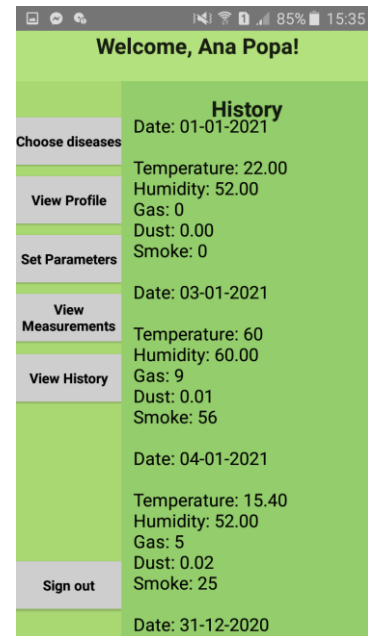


Figura 15: Vizualizarea
istoricului

Vizualizarea valorilor masurate

În funcție de parametrii pe care i-am ales, putem să vedem valorile pentru fiecare parametru, după acționarea butonului de “View measurements” din partea stângă a ecranului. Acest buton o să ne redirecționeze către o pagină ca cea din Figura 16. Pentru a prelua datele de la senzori trebuie să ne conectăm prin intermediul bluetooth-ului la placuta arduino. Acest lucru se face prin acționarea butonului “Connect” din partea dreapta-sus a ecranului. Dacă avem bluetooth-ul deja activ o să ne apară direct lista cu dispozitivele imperecheate, dacă nu avem bluetooth-ul activ, o să ne apară o fereastră de dialog prin care ni se cere acordul în legătură cu activarea bluetooth-ului (Figura 17). După ce ne-am dat acordul, o să trebuiască să apăsăm “ok” pe fereastra de dialog din partea de jos a ecranului pentru ca dispozitivele să apară (Figura 18).

Dacă conectarea nu a reușit, o să fim avertizați printr-un mesaj în partea de sus a ecranului (Figura 20), iar dacă am reușit să ne conectăm la dispozitivul selectat din lista (Figura 19), o să ne apară numele dispozitivului (Figura 21). Senzorii transmit datele la senzori în mod continuu, astfel încât datele sunt mereu actualizate. Dacă totuși vrem ca pagina noastră să se reîncarce, avem la dispoziție butonul de “Refresh”. Dacă vrem să ne întoarcem la pagina anterioară, trebuie doar să apăsăm butonul “Return to main page”.

Figura din partea de sus a ecranului o să indice culoarea galben la start, culoarea verde dacă totul e în regulă și culoarea roșie dacă există vreun pericol care să ne afecteze. În funcție de afecțiunile pe care le-am selectat, o să fim avertizați dacă ceva este în neregulă printr-un mesaj.

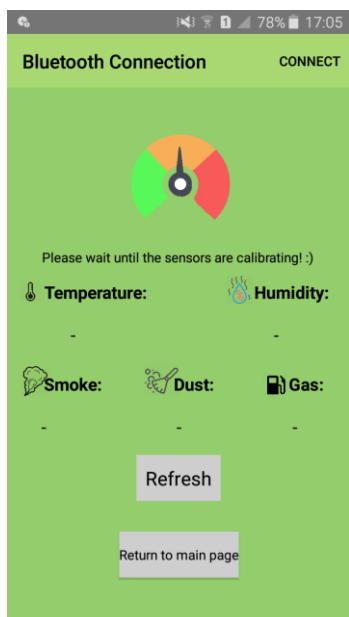


Figura 16: Vizualizarea
masuratorilor

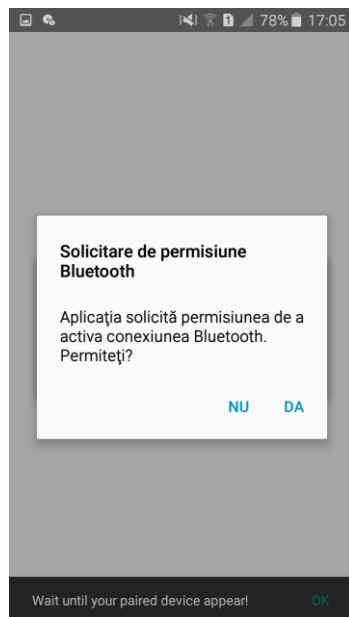


Figura 17: Activarea
bluetooth-ului

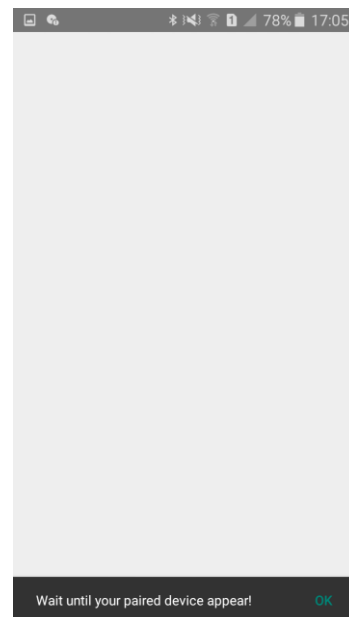


Figura 18: Așteptarea pentru
dispozitivele imperecheate



Figura 19: Dispozitivele
imperecheate

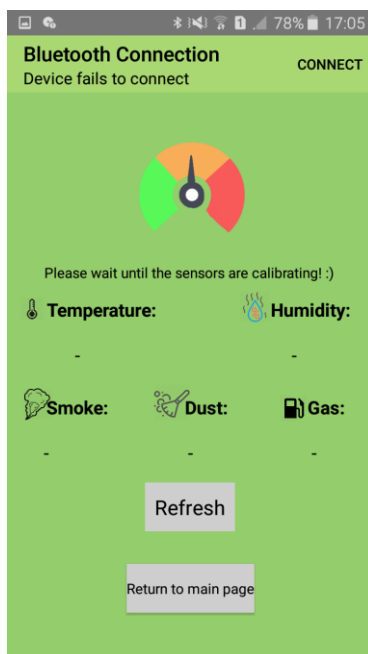


Figura 20: Eroarea la
conectare

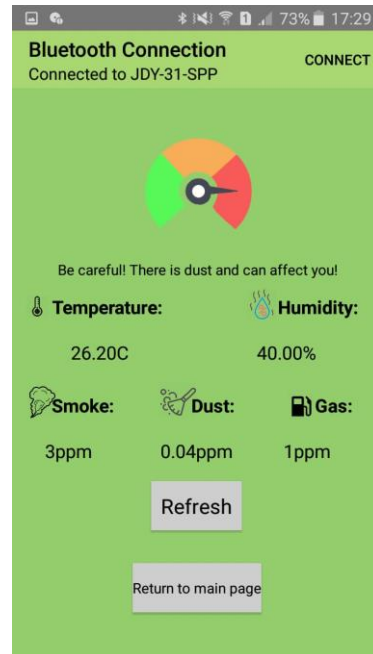


Figura 21: Afisarea datelor
primate de la senzori

Pagina de administrator

Ca si administrator, putem sa ne logam si sa vedem informatii despre toti utilizatorii care si-au creat un cont, dar putem sa vedem si persoanele care mai sunt administratori, cu exceptia noastra.

Butonul de Sign-out ne permite sa ne deconectam.

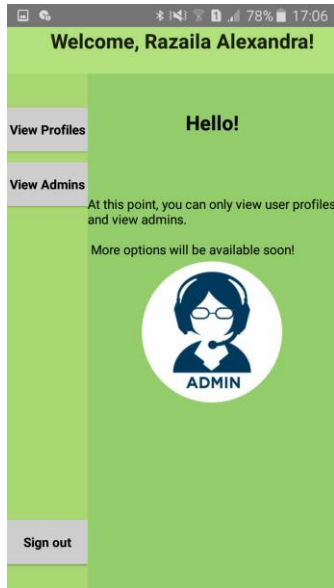


Figura 22: Pagina de administrator

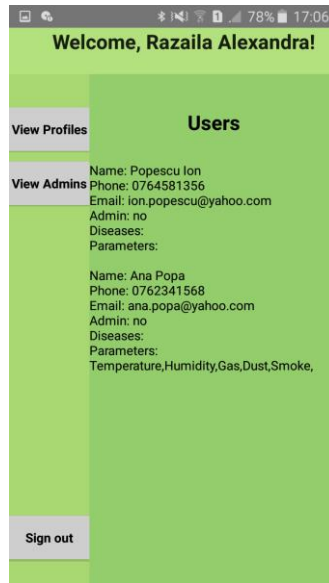


Figura 23: Afisarea utilizatorilor

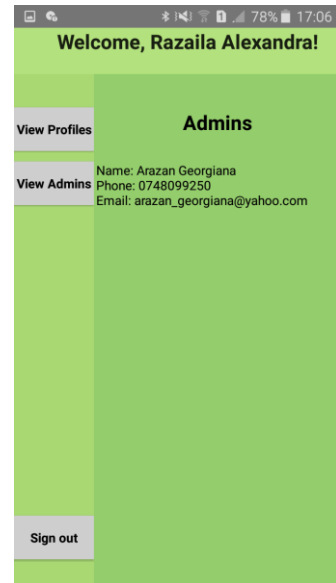


Figura 24: Afisarea administratorilor

Concluzii

Dupa cum am spus si in introducere, proiectul nostru vine in ajutorul celor care sufera de astm sau de alergii la praf pentru a-i ajuta sa descopere ce anume din aer reprezinta un pericol pentru ei masurand parametrii precum temperatura, umiditatea, cantitatea de gaz, fum si praf din aer si atentioneaza utilizatorii printr-un mesaj in cazul in care ceva nu este in regula.

Principalele lucruri pe care as vrea sa le schimb sau sa le adaug aplicatiei sunt:

- Sa nu mai fie hard-codate denumirile din aplicatie(string-urile pentru interfata) pentru a putea sa o fac sa fie disponibila si in alta limba.
- Administratorul sa poata sa schimbe drepturile de admin si informatiile legate de bolile din baza de date.
- Sa folosesc un senzor dht22 pentru a putea masura temperatura sub 0 grade Celsius.

Bibliografie

1. <https://en.wikipedia.org/wiki/GitHub>
2. <https://en.wikipedia.org/wiki/Arduino>
3. <https://prezi.com/p/mtkhq9poukw/firebase/>
4. <https://roboromania.ro/2016/11/15/descrierea-pinilor-la-placa-arduino-uno-r3/>
5. [https://ro.wikipedia.org/wiki/Java_\(limbaj_de_programare\)](https://ro.wikipedia.org/wiki/Java_(limbaj_de_programare))
6. https://ro.wikipedia.org/wiki/Android_Studio
7. <https://www.adafruit.com/product/386>
8. <https://www.tme.eu/Document/d08daa6b6b1f3ee103aa929b49cff2ca/GP2Y1010AU0F-en.pdf>
9. <https://sites.google.com/site/computerulraspberrypi/componente/1-temperature-humidity-air-pressure-gas/senzor-de-gaz-mq-2>
10. <https://www.sigmanortec.ro/Modul-Bluetooth-BT-06-p134711942>