

# **PlugTim**

## **Aplicatie pentru gestionarea statiilor de incarcare a masinilor electrice**

**Student:**

**Dumitrescu Bogdan-Doru**

**Timisoara**

**Ianuarie, 2022**

## **Cuprins:**

- 1. Introducere**
- 2. State of the art**
- 3. Tehnologii folosite**
- 4. Proiectarea sistemului**
- 5. Implementarea functionalitatilor**
- 6. Utilizarea aplicatiei**
- 7. Monitorizarea calitatii**

# Capitolul 1

## Introducere

### 1.1 Context

Anul 2021 marcheaza cea de-a 26-a Conferinta a partilor (COP26) la Conventia Natiunilor Unite asupra schimbarilor climatice, si cea mai importanta de pana acum. Potrivit National Climate Assessment and Development Advisory Committee, infiintat in 2010 de Departamentul American al Comertului cresterea temperaturii medii poate depasi 5 grade Celsius pana in 2100. Previziunile optimiste, care se bazeaza pe o reducere substantiala a acestor emisii pana in 2050, arata o crestere a temperaturii medii intre 1,6 si 2,7 pana la sfarsitul secolului. Membrii comitetului amintesc ca din 1895, temperature medie a crescut cu doar 0,8 grade, dintre care 80% din 1980 si afirma ca schimbarea climatica rapida e un semnal de avertizare.

Tot in 2021, in Romania, s-a inregistrat o crestere substantiala a inmatricularilor de masini electrice. Numarul acestora a depasit totalul inmatricularilor realizate intre anii 2011 si 2019. Aceasta crestere m-a facut sa imi pun doua intrebari importante: care este numarul de statii de incarcare si unde se afla acestea. Posibil cel mai mare dezavantaj al unei masini electrice este chiar cea mai mare calitate a acesteia, bateria. Bateria unei masini electrice este de tip li-ion, aceeași folosita la smartphone-uri sau laptopuri, dar de dimensiunii mai mari. Totusi, aceasta baterie are o durata de viata foarte scurta, iar locatia unei statii de incarcare in apropierea soferului devine o nevoie importanta.

Deoarece nu exista o baza de date concreta pentru statii de incarcare, mi-a venit ideea de a creea o aplicatie care va distribui locatiile statilor de incarcare dintr-un anumit oras introduse in aplicatie chiar de utilizatorii ei.

## **1.2 Descrierea proiectului**

Ideea principală a aplicației “PlugTim” este de a oferi utilizatorului locația unei stații de încărcare din apropiere, dar și șansa de a ajuta un alt utilizator prin a adăuga o stație nouă descoperită de acesta.

Aplicația “PlugTim” ajută orice locuitor al orașului Timisoara, care este deținător al unui vehicul electric. Aceasta reduce timpul de căutare a unei stații în fizic, printr-o simplă deschidere a aplicației.

După crearea rapidă a unui cont, aplicația permite utilizatorului o hartă intuitivă, pe care se pot vedea locațiile stațiilor de încărcare. La nevoie, aceasta poate fi folosită o aplicație de hărți pentru a ajunge la stația dorită.

Opțiunea de a adăuga o stație este foarte intuitivă, aplicația obținând, după permisiunea utilizatorului, locația stației noi adăugate și șansa de a oferi mai multe detalii despre stație printr-un formular scurt. În plus, va exista și opțiunea de a fotografia stația.

În final, aplicația va putea oferi utilizatorului informațiile adăugate de acesta, iar dacă a ales să adăuge o mașină, să vadă informațiile despre aceasta.

## Capitolul 2

### Analiza domeniului

#### 2.1 Aplicatii asemanatoare

În urma unei căutări pe Magazin Play, aplicație oferită de Google, principala aplicație descărcată de utilizatori de Android, cu o recenzie de 4,4 din 5 și peste 1 milion de descărcări, este PlugShare. Putem observa diverse asemănări între cele 2 aplicații prin funcționalitățile oferite.

Principalele funcționalități oferite de ambele aplicații sunt:

- **crearea unui cont și adăugarea unei mașini**
- **o hartă cu toate stațiile de încărcare din apropierea utilizatorului**
- **gestionarea informațiilor personale adăugate în cont**

O diferență este posibilitatea de a crea o rută de călătorie în cadrul aplicației PlugShare, în timp ce aplicația dezvoltată de mine nu are această funcționalitate, aceasta fiind dezvoltată strict pentru un anumit oraș. O altă diferență este oferirea posibilității utilizatorului de a adăuga o stație de încărcare în cadrul aplicației dezvoltată de mine, dar una care este dificil de verificat.

Următoarea aplicație din listă este numită Chargemap, cu o recenzie de 4,3 din 5 și un număr de descărcări ce depășește 500000. Aceasta este asemănătoare cu cele 2 aplicații descrise mai sus, dar având o asemănare cu aplicația dezvoltată de mine prin posibilitatea de a adăuga stații de încărcare. O diferență între aceasta și cele descrise anterior este legată de comunitate. Aceasta aplicație oferă posibilitatea de “check-in” prin care poți descrie o încărcare la o anumită stație.

# **Capitolul 3**

## **Tehnologii folosite**

### **3.1 Autentificare**

#### **3.1.1 Firebase Authentication with email and password**

#### **3.1.2 Firebase Email address verification**

#### **3.1.3 Firebase Password Reset**

### **3.2 Baza de date**

#### **3.2.1 Firebase Realtime Database**

#### **3.2.2 Firebase Cloud Storage**

### **3.3 Java**

### **3.4 Android**

### **3.5 Android Studio**

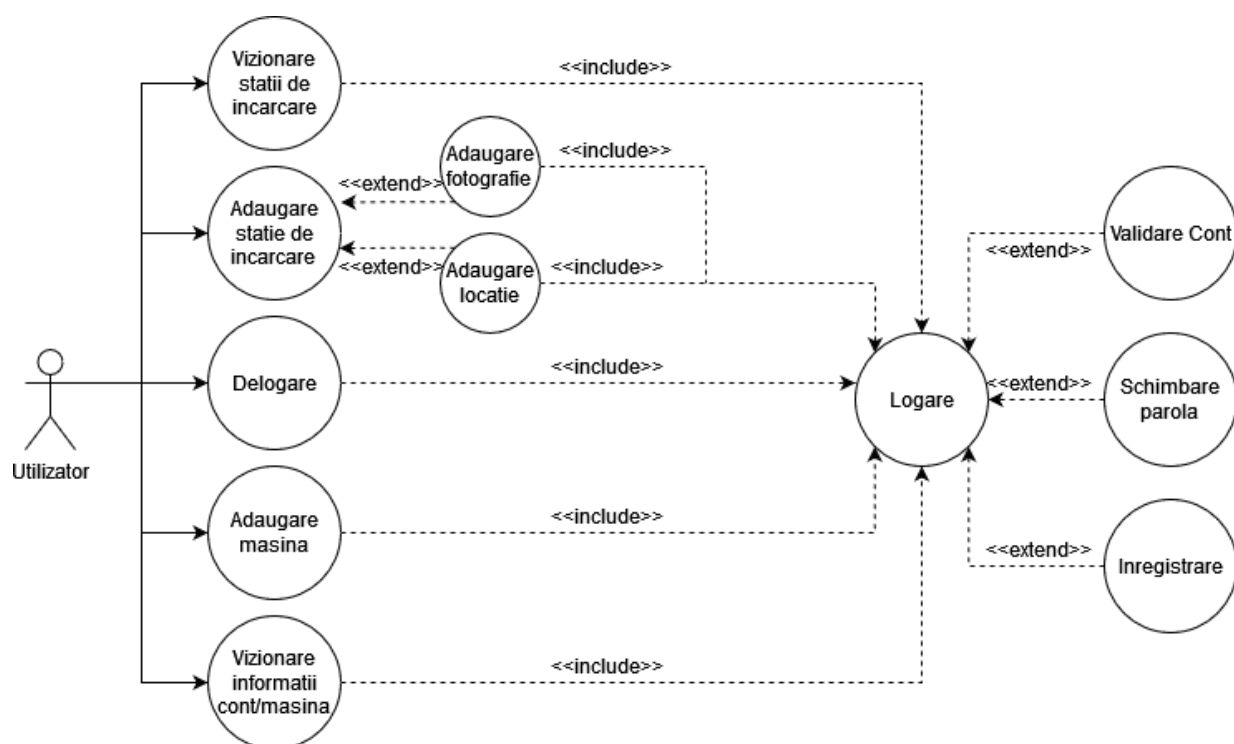
### **3.6 Versionarea codului – Git**

### **3.7 Maps SDK for Android**

# Capitolul 4

## Proiectarea sistemului

### 4.1 Diagrama UML a cazurilor de utilizare



# Capitolul 5

## Implementarea functionalitatilor

### 5.1 Inregistrarea

Inregistrarea unui utilizator in aplicatie este formata din doua parti: inregistrarea unor date personale constituie prima parte obligatorie, iar inregistrarea masinii utilizatorului, daca acesta doreste, este a doua parte.

Prima parte consta in crearea unui cont prin email si parola, folosind Firebase Authentication, si oferirea unui prenume si nume care sunt stocate impreuna cu email in Firebase Realtime Database, la o locatie recunoscuta prin uuid-ul userului. Inainte de a trece la pasul de inregistrare a masinii se fac verificariile necesare, precum completarea campurilor, oferirea unui email valid, o parola care sa contina cel putin 6 caractere

```
if (username.isEmpty()) {
    regUsername.setError("Email is required");
    regUsername.requestFocus();
    return false;
} else if (!Patterns.EMAIL_ADDRESS.matcher(username).matches()) {
    regUsername.setError("Email is invalid");
    regUsername.requestFocus();
    return false;
} else if (password.isEmpty()) {
    regPassword.setError("Password is required");
    regPassword.requestFocus();
    return false;
} else if (password.length() < 6) {
    regPassword.setError("Password must have at least 6 characters");
    regPassword.requestFocus();
    return false;
} else if (firstName.isEmpty()) {
    regFirstName.setError("First Name is required");
    regFirstName.requestFocus();
    return false;
} else if (lastName.isEmpty()) {
    regLastName.setError("Last Name is required");
    regLastName.requestFocus();
    return false;
}
```

Validarea campurilor



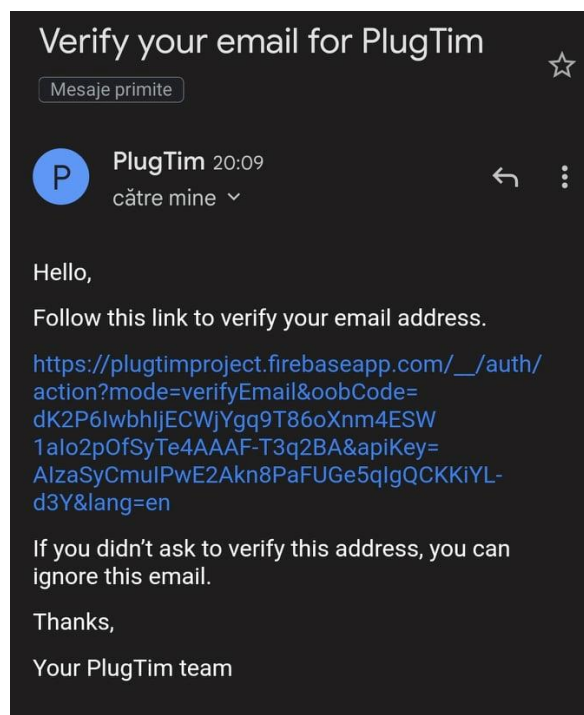
Dupa ce campurile au fost validate, se creeaza un utilizator nou folosind o metoda predefinita **createUserWithEmailAndPassword(email, password)**. Daca utilizatorul a fost create cu success, atunci se apeleaza metoda **sendEmailVerification()**. Aceasta trimite un email la adresa de email oferita, iar utilizatorul va putea intra in aplicatie cu contul nou creat, doar dupa verificare.

```
mAuth = FirebaseAuth.getInstance();
mDatabase = FirebaseDatabase.getInstance().getReference();

mAuth.createUserWithEmailAndPassword(username, password)
    .addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            User user = new User(username, firstName, lastName);

mDatabase.child("users").child(Objects.requireNonNull(mAuth.getCurrentUser()
)).getUid())
            .setValue(user).addOnCompleteListener(task1 -> {
                if (task1.isSuccessful()) {
                    FirebaseUser firebaseUser =
FirebaseAuth.getInstance().getCurrentUser();
                    firebaseUser.sendEmailVerification();
                    Toast.makeText(Register.this, "A verification email
has been sent. Please check your email", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(Register.this, "Failed to register.
Please try again", Toast.LENGTH_SHORT).show();
                }
            });
        }
    });
```

### Creearea unui utilizator



De asemenea, exista si optiunea de resetare a parolei prin email. Pentru aceasta s-a folosit metoda predefinita **sendPasswordResetEmail()**. Inainte de a se apela, se verifica daca un email valid s-a introdus in campul corespunzator,

## 5.2 Adaugarea unei statii de incarcare si afisarea acesteia in activatea Google Maps

Pentru a putea folosi o activitate de tip Google Maps, in aplicatie s-a utilizat Maps SDK. Un utilizator poate adauga o statie de incarcare, doar daca a adaugat o descriere si numarul de porturi de incarcare. Dupa verificarea acestor campuri, acesta are optiunea de a adauga o poza la statie de incarcare. Aceasta va fi incarcata in Firebase Cloud Storage. Informatiile statiei, cat si latitudinea si longitudinea acesteia, vor fi stocate in Realtime Database la un uuid generat, atribuit si pozei. Locatia statiei este considerata ca fiind locatia curenta a utilizatorului. Acesta va permite accesarea locatiei sale, iar apoi aceasta va fi stocata impreuna cu statia.

```
if (ActivityCompat.checkSelfPermission(AddStation.this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
    locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

    if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {
        fusedLocationClient.getLastLocation().addOnCompleteListener(task ->
{
            Location location = task.getResult();
            if (location != null) {
                String descriptionStation =
description.getText().toString();
                String portsStation = ports.getText().toString();
                Geocoder geocoder = new Geocoder(AddStation.this,
Locale.getDefault());
                try {
                    List<Address> addressList =
geocoder.getFromLocation(location.getLatitude(), location.getLongitude(),
1);

                    Station station = new Station(descriptionStation,
Integer.parseInt(portsStation), new
LocationHelper(addressList.get(0).getLatitude(),
addressList.get(0).getLongitude()), FirebaseAuth.getInstance().getUid());

                    String random = UUID.randomUUID().toString();

                    mDatabase.child("stations").child(random)
                        .setValue(station)
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

Adaugarea unei statii

In MapsActivity, activitatea de tip Google Maps, este folosit un clicklistener pentru markere, care va afisa un popup cu informatiile statiei, cat si o poza a acesteia.

```
dialogBuilder.setView(mapsPopupView);
dialog = dialogBuilder.create();
dialog.show();

// add photo in dialog

mDatabase.child("stations").addValueEventListener(new ValueEventListener()
{
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        stationList.clear();
        uuids.clear();
        for (DataSnapshot postSnapshot : snapshot.getChildren()) {
            Station station = postSnapshot.getValue(Station.class);
            stationList.add(station);
            uuids.add(postSnapshot.getKey());
        }

        for (int i = 0; i < stationList.size(); i++) {
            if (stationList.get(i).getLocationHelper() != null) {
                String description = "";
                String ports = "";
                StorageReference pathReference =
storageReference.child("images/" + uuids.get(i));

                if (stationList.get(i).getLocationHelper().getLatitude() ==
latitude && stationList.get(i).getLocationHelper().getLongitude() ==
longitude) {
                    description = stationList.get(i).getDescription();
                    ports =
stationList.get(i).getNumberOfPorts().toString();

                    long MAXBYTES = 1024 * 1024 * 20;

                    pathReference.getBytes(MAXBYTES).addOnSuccessListener(new
OnSuccessListener<byte[]>() {
                        @Override
                        public void onSuccess(@NonNull byte[] bytes) {
                            Bitmap bitmap =
BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
                            evPhoto.setImageBitmap(bitmap);
                        }
                    });

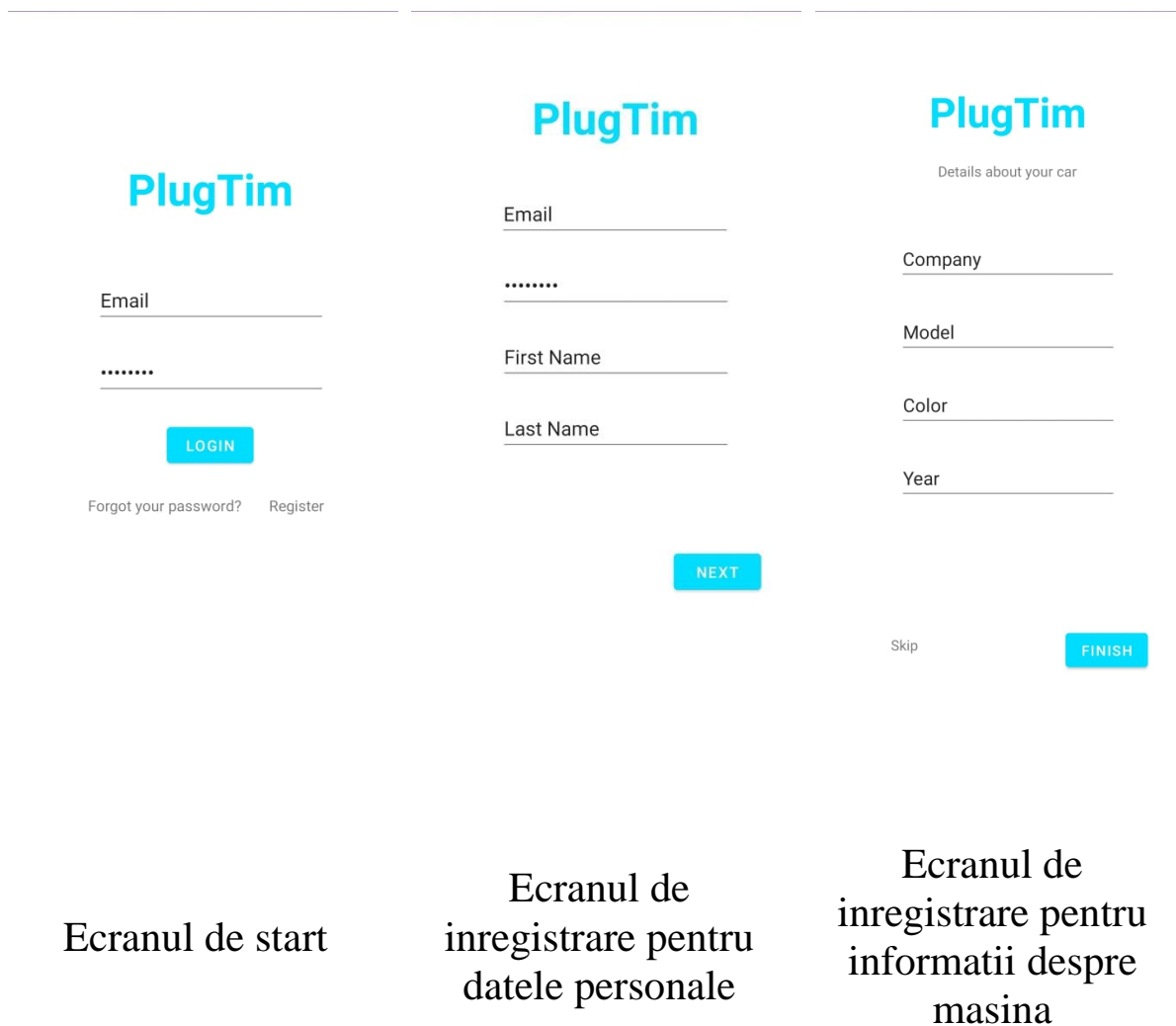
                    descriptionMaps.setText(description);
                    portsMaps.setText(ports);
                }
            }
        }
    }
});
```

Adaugarea pozei si informatiei in popup

# Capitolul 6

## Utilizarea aplicatiei

### 6.1 Inregistrare/Logare/Resetare Parola



---

**PlugTim**

Email

RESET

Ecranul de resetare a parolei

## 6.2 Informatii cont si masina

---

**PlugTim**

Email:

@gmail.com

First Name:

Bogdan

Last Name:

Dumitrescu

**PlugTim**

Car Company:

Dacia

Car Model:

Logan

Color:

Grey

Year

2022

Ecranul cu informatii  
despre cont

Ecran cu informatii despre  
masina


## 6.3 Adaugarea statie

### PlugTim

Please enter station's details

Statie EnelX

2



UPLOAD PHOTO

ADD STATION

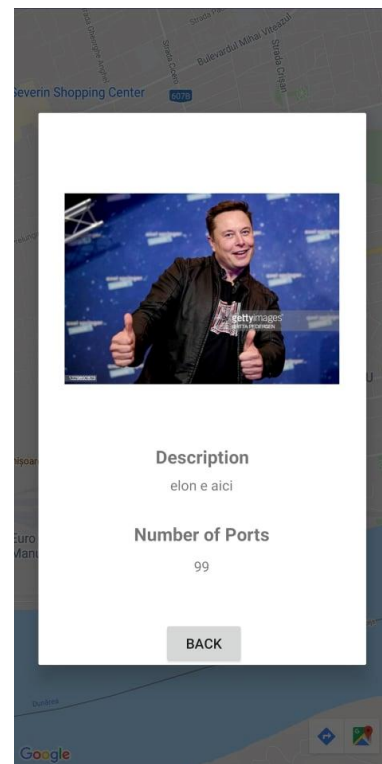
To complete this operation you will need to give access to your location

Ecran de adaugare a statiei

## 6.4 Vizionare statii



Ecran cu harta



Ecran cu informatii

## Capitolul 7

### Monitorizarea calitatii

Nota pe care as acorda-o proiectului si muncii mele este 8, deoarece proiectul este departe de a fi finalizat. Exista multe lucruri pe care as dori sa le schimb la el si sper sa am ocazia cand o sa il continui la licenta. Tinand cont de ideile de la inceput, proiectul le contine pe toate, chiar daca nu sunt implementate in cel mai corect mod.

Linkul catre repository este:

<https://github.com/DumitrescuBogdanDoru/PlugTimProject.git>

# Bibliografie

[1] Documentie Android

<https://developer.android.com/docs/>

[2] Documentie Firebase

<https://firebase.google.com/docs?authuser=1>

[3] Documentatie Maps SDK

<https://developers.google.com/maps/documentation/android-sdk/overview>

[4] Iuliana Enache, Incalzirea globala, o realitate.

<https://www.mediafax.ro/stiinta-sanatate/incalzirea-globala-o-realitate-temperatura-ar-putea-creste-cu-5-grade-celsius-pana-in-2100-10456219>

[5] Wall-Street, Numarul de masini electrice in 2021 a depasit totalul inmatricularilor realizate intre anii 2011 si 2019

<https://www.wall-street.ro/articol/Auto/278491/numarul-de-masini-electrice-inmatriculate-in-2021-a-depasit-totalul-inmatricularilor-realizate-intre-anii-2011-si-2019.html>