



Universitatea *Transilvania* din Brașov
Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul Automatică și Tehnologia Informației



PROIECT LA SDA - JAVA

APLICATIE PARCARE COLINA



Apetrei Diana-Alexandra
Gr. 4891, ROBO II

Balog Annamária
Epure Alexandru - Cristian
Găitănaru Mihaela - Iulia
Gr. 4391, TI II

BRAȘOV, 2021

Cuprins

1. Aspecte generale	3
1.1. Tema aplicației	3
1.2. Funcționarea generală a aplicației	3
1.3. Riscuri	3
1.4. Prima interacțiune cu clientul.....	3
1.5. Android studio.....	3
1.6. Firebase	4
2. Pagina de logare	5
2.1. Pagina de ajutor	5
2.2. Resetare parolă	5
2.3. Logare	6
3. Meniu Studenți.....	7
3.1. Rezervare.....	7
3.2. Status	8
3.3. Istoric parcări	8
3.4. Raportează.....	8
3.5. Hartă parcare	9
3.6. Profilul meu.....	9
3.7. Elemente de aspect.....	10
3.8. Sortare în baza de date	10
4. Meniu admini	11
4.1. Firebase Admin WEB	11
4.2. Adaugare utilizatori.....	12
4.3. Vizualizare membrii.....	12
4.4. Modifică membrii.....	13
4.5. Analizare reporturi	13
4.6. Zona Admin	13
4.7. Contul meu	13
4.8. Resetare parola	13
4.9. Butoane adiționale.....	14
4.10. Algoritm de căutare.....	14
5. Concluzii	15
5.1. Concluzii generale.....	15
5.2. Elemente de originalitate.....	15
5.3. Direcții viitoare de dezvoltare	15
5.4. Impresii	15
6. Bibliografia	16

1. Aspecte generale

1.1. Tema aplicației

Cum digitalizarea și tehnologizarea apar în aproape toate aspectele vieții omenești și cum mașinile aproape au devenit computere pe patru roți, era inevitabil ca cineva să nu se gândească la o aplicație care ajută la găsirea unui loc de parcare și care spune și unde ai lăsat autoturismul. Astfel, proiectul nostru are ca scop realizarea unei aplicații Android, care să funcționeze ca un “Manager al locurilor de parcare”.

1.2. Funcționarea generală a aplicației

Prin intermediul aplicației, un șofer este informat care locuri de parcare sunt libere într-o anumită zonă și poate rezerva un anumit loc. Aplicația va funcționa pe baza unui Sistem de LOGIN, iar procesul de autentificare în aplicație se va face prin intermediul unui cont instituțional. De asemenea, aplicația pune la dispoziție utilizatorului sisteme pentru: gestionarea numerelor de înmatriculare; raportarea neclarităților și a locurilor de parcare ocupate în mod necorespunzător; raportarea urgențelor; afișarea în timp real a locurilor de parcare ocupate, precum și numerele de înmatriculare ale autovehiculelor aferente; resetare a parolei contului. Majoritatea datelor vor fi stocate într-o bază de date.

1.3. Riscuri

S-au analizat cerințele proiectului și s-au identificat următoarele riscuri:

- Necunoașterea tehnologiilor ce vor fi necesare în rezolvarea proiectului;
- Risc major din cauza lipsei comunicării cu clientul pe parcursul proiectului;
- Nerespectarea termenelor proiectului;
- Situația pandemică nu ne-a permis să dezvoltăm aplicația într-un mediu fizic.

1.4. Prima interacțiune cu clientul

Prima noastră interacțiune cu clientul a fost pe data de 22.10.2020, în care i-am propus acestuia ideea proiectului nostru, iar acesta a acceptat sugestia.

1.5. Android studio



Fig. 1.5. Logo Android Studio

Android Studio este un mediu de dezvoltare (engl. software development environment, sau integrated development environment - "mediu integrat de dezvoltare) pentru colaborarea cu platforma Android, anunțată pe data de 16 mai 2013 în cadrul conferinței I / O Google.

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. Pe 17 mai 2017, la conferința anuală Google I / O, Google a anunțat asistență pentru limbajul Kotlin utilizat de Android Studio ca limbaj de programare oficial pentru platforma Android, pe lângă Java și C ++.

Am ales utilizarea acestui IDE (mediu de dezvoltare integrat) datorită eficacității sale.

1.6. Firebase



Fig. 1.2. Logo Firebase

Firebase este o platformă de dezvoltare pentru aplicații mobile și web. Cu Firebase, puteți să creați rapid aplicații de înaltă calitate, să stabiliți o bază de utilizatori implicată și să câștigați mai mulți bani. Platforma include mai multe funcții strâns legate între ele, pe care le puteți combina după cum doriți. Printre acestea se numără soluții de analiză, un backend mobil și

instrumente de dezvoltare a aplicațiilor și de generare de venituri, prin care să maximizați succesul aplicației.

Realtime database este un sistem de baze de date care utilizează procesarea în timp real pentru a gestiona sarcini de lucru a căror stare se schimbă constant. Acest lucru diferă de bazele de date tradiționale care conțin date persistente, în mare parte neafectate de timp.

Firestore este o bază de date de documente NoSQL construită pentru scalare automată, performanță ridicată și ușurință în dezvoltarea aplicațiilor. În timp ce interfața Firestore are multe dintre aceleași caracteristici ca bazele de date tradiționale, ca bază de date NoSQL diferă de acestea prin modul în care descrie relațiile dintre obiectele de date.

2. Pagina de logare

2.1. Pagina de ajutor

Scopul paginii de ajutor este de a oferi informații referitoare la funcționarea aplicației studenților ce o folosesc.

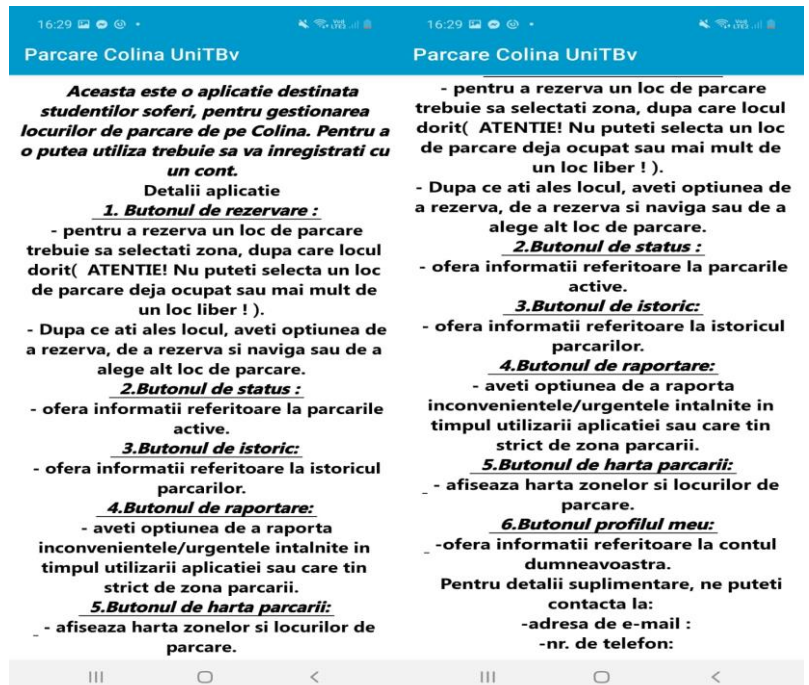
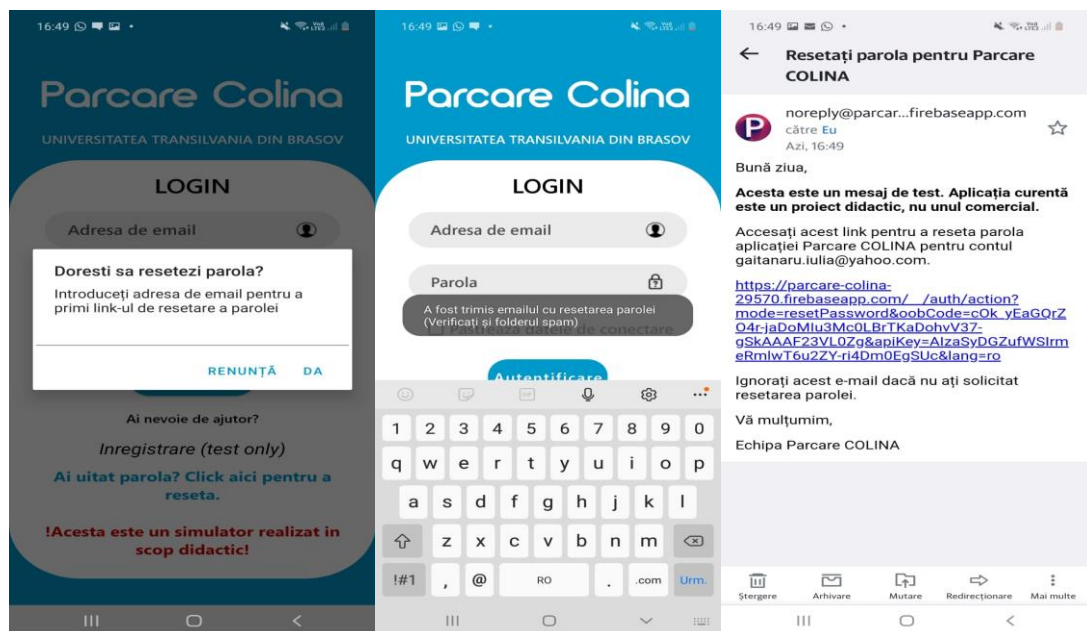


Fig. 2.1. Reprezintă modul de afișare al paginii de ajutor

2.2. Resetare parolă

Oferă posibilitatea de resetare a parolei în cazul în care utilizatorul a uitat-o.

După introducerea adresei de email asociată contului, se va trimite automat prin intermediul aplicației un mail către utilizator care conține indicațiile cu privire la procesul de resetare a parolei.



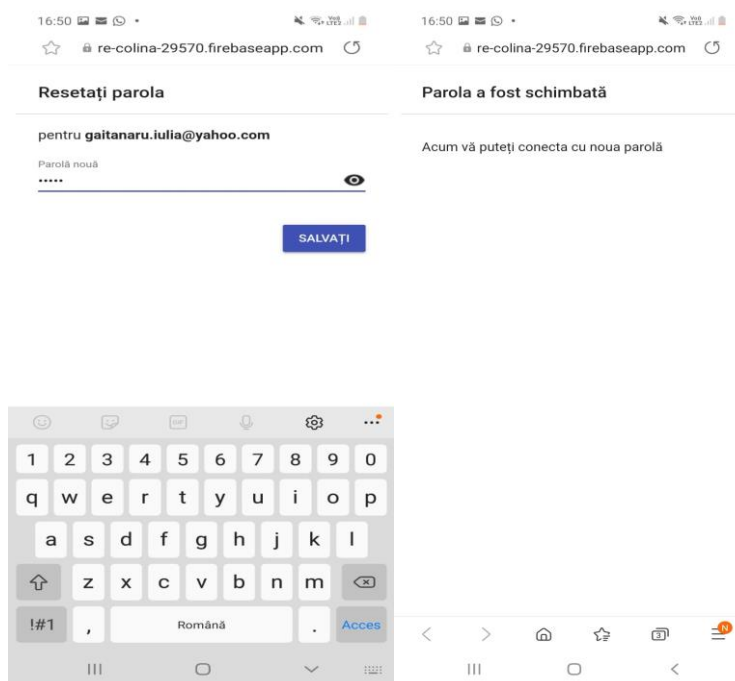


Fig. 2.2. Reprezintă pașii pentru resetarea parolei



Fig. 2.3. Reprezintă pagina de
logare

2.3. Logare

Dupa ce se crează contul , vă puteți loga cu un cont de student sau de admin, în funcție de tipul contului existent. După logare, veți fi trimis în meniul principal de student sau de admin. De asemenea, se pot păstra datele de conectare.

3. Meniu Studenți

Meniul Studenților cuprinde următoarele utilități:

- Rezervarea unui loc de parcare
- Vizualizarea statusului parcării
- Istoricul parcărilor
- Raportarea urgențelor
- Vizualizarea hărții parcării
- Vizualizarea profilului personal
- Pagina de ajutor



Fig. 3. Meniul studenților

3.1. Rezervare

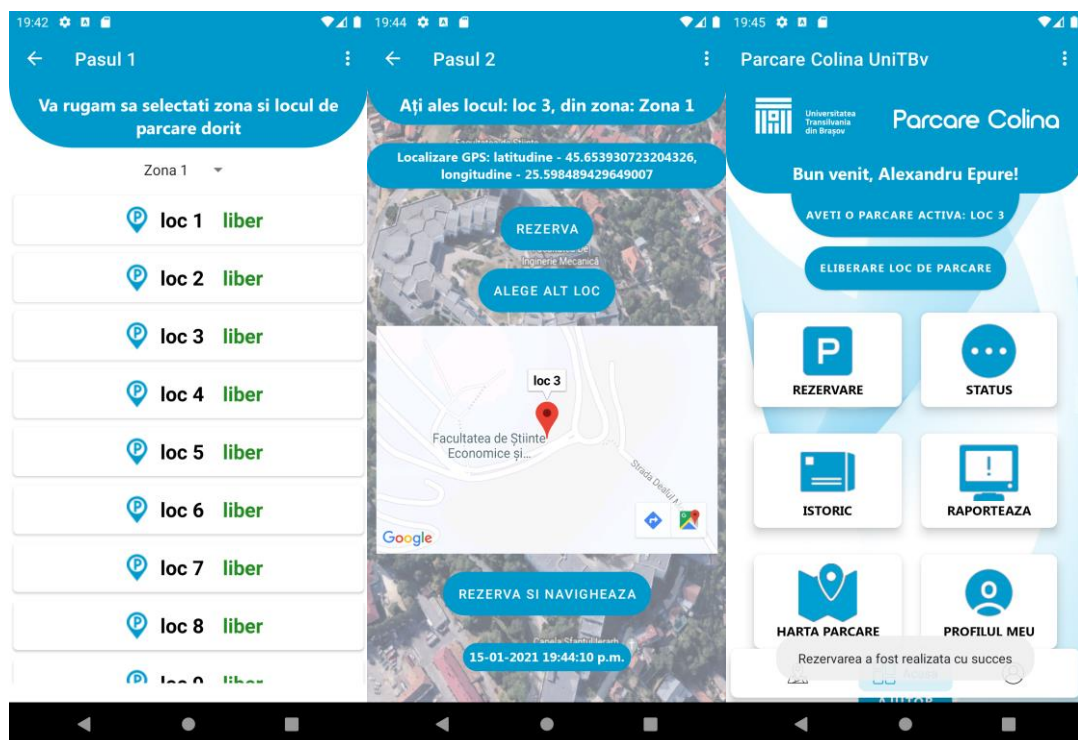


Fig. 3.1. Rezervarea locului de parcare

Scopul acestei utilități este de a rezerva un loc de parcare. Pașii care trebuie urmați sunt:

- Alegerea zonei și a locului dorit de parcare
- Rezervă/ Alegere alt loc/ Rezervă și navighează
- Odată ce ați rezervat locul dorit, se va afișa o parcare activă
- Când doriți să părăsiți locul de parcare, aveți datoria de a-l elimina.

3.2. Status

Afișează parcare activă cu opțiunea de a vizualiza istoricul parcărilor și de a elibera locul de parcare activ.

3.3. Istoric parări

Afișează locurile de parcare anterioare.

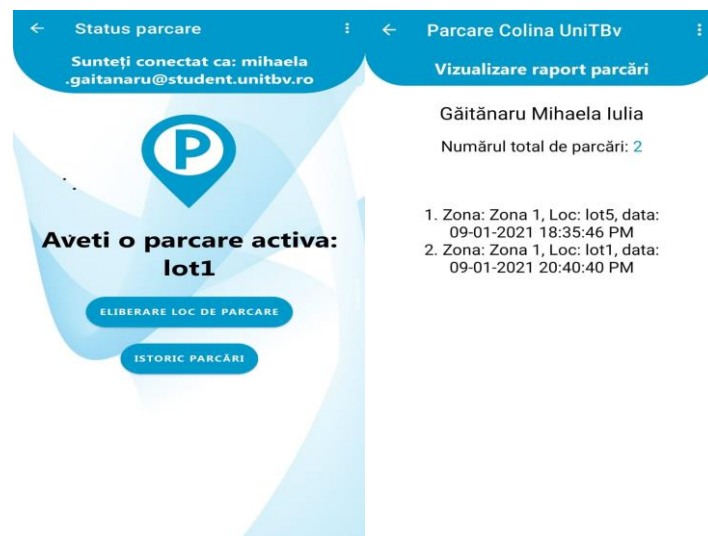


Fig. 3.2. Status parcare

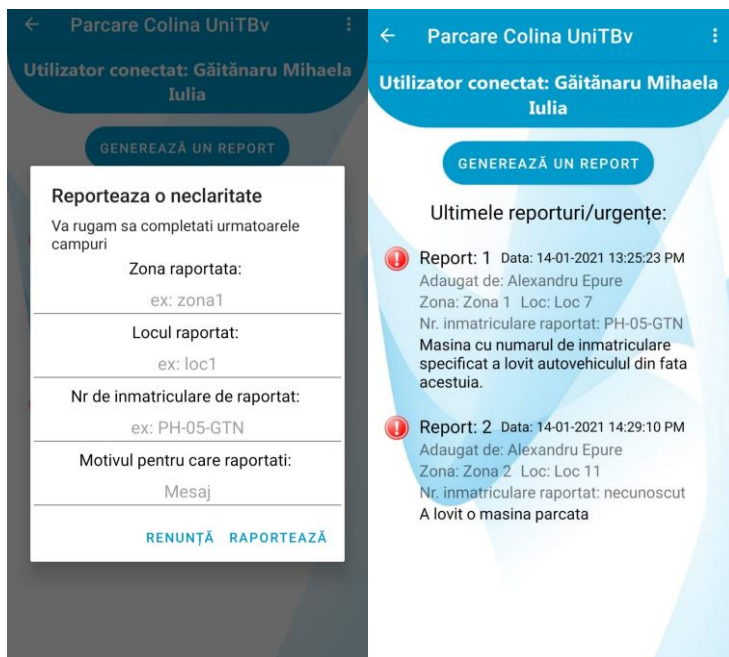


Fig. 3.4.1. Generarea raportului Fig. 3.4.2. Afișarea raporturilor

3.4. Raportează

Se poate genera un raport cu privire la diferitele urgențe, urmând să fie afișat.

În raport trebuie menționate:

- Zona și locul raportat ;
- Nr. de înmatriculare raportat ;
- Motivul pentru care raportați .

3.5. Hartă parcare

Afișează harta cu toate locurile de parcare.

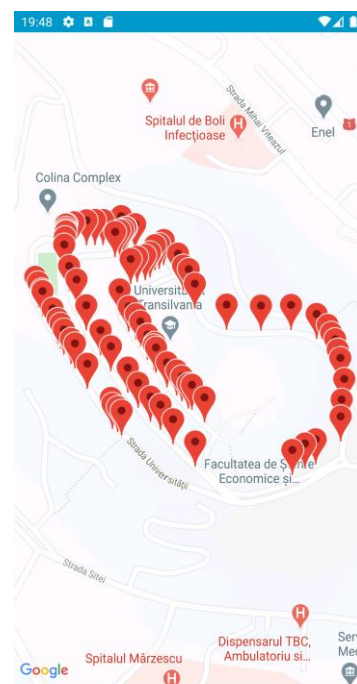


Fig. 3.5. Harta parcării

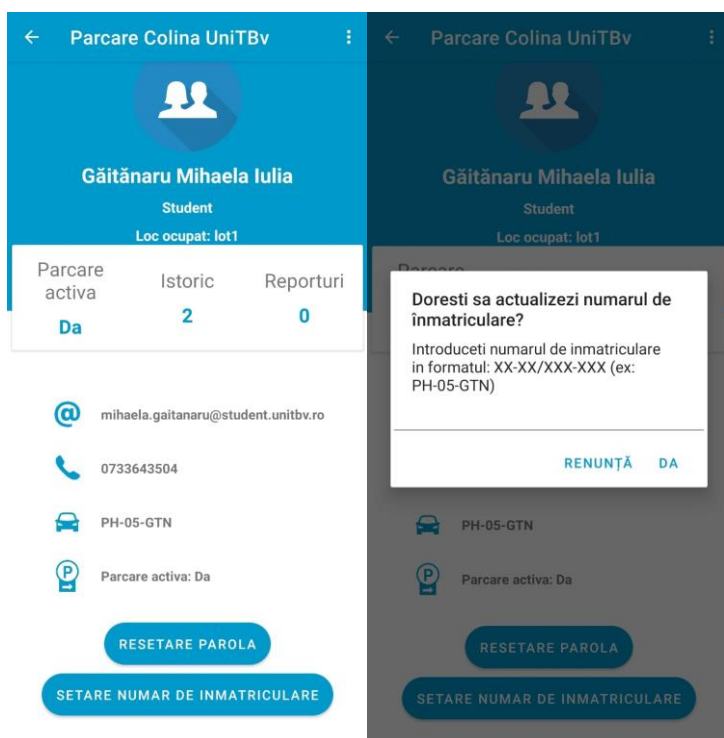


Fig. 3.5.1. Profilul meu

Fig. 3.5.2. Resetare nr. de înmatriculare

3.6. Profilul meu

Afișează informații generale ale utilizatorului. (nr. de telefon, nr. de înmatriculare, istoric parcări, reporturi, parcare activă, mail, nume etc.)

Se poate reseta parola sau nr. de înmatriculare.

3.7. Elemente de aspect

a. Navigation Bar

Conține 3 elemente :

- Harta parcării ;
- Acasă (Meniul principal) ;
- Profilul utilizatorului .

b. Meniu în bara de acțiune

Conține 4 elemente :

- Acasă (Meniul principal) ;
- Ajutor ;
- Profilul utilizatorului ;
- Delogare.



Fig. 3.7.a. Navigation Bar



Fig. 3.7.b. Meniu în bara de acțiune

3.8. Sortare în baza de date

Clasa Query (și subclasa sa, DatabaseReference) sunt utilizate pentru citirea datelor. Ascultătorii sunt atașați și vor fi declanșați la schimbarea datelor corespunzătoare.

Instanțele de interogare sunt obținute apelând startAt (), endAt () sau limit () pe un DatabaseReference.

Pentru a prelua date sortate, începeți prin specificarea uneia dintre metodele de ordonare pentru a determina modul în care sunt ordonate rezultatele:

Method	Usage
orderByChild()	Order results by the value of a specified child key or nested child path.
orderByKey()	Order results by child keys.
orderByValue()	Order results by child values.


```

Query query = rfl.orderByChild("pos");
recview.setLayoutManager(new
LinearLayoutManager(this));
FirebaseRecyclerOptions<ModelLot> options =
    new
    FirebaseRecyclerOptions.Builder<ModelLot>()
        .setQuery(query,
ModelLot.class)
        .build();
adapter=new AdapterRecyclerView(options);
recview.setAdapter(adapter);
    
```

4. Meniu admini



Meniul de admini cuprinde următoarele opțiuni:

- Firebase Admin WEB
- Adăugare utilizatori
- Vizualizare membrii
- Modifică membrii
- Analizare raporturi
- Zona Admin
- Contul meu
- Resetare parolă
- Butoane adiționale

Fig 4. Meniu admini

4.1. Firebase Admin WEB

Din Admin Panel Web se pot accesa bazele de date, se pot vedea statistici si se pot adauga/modifica date

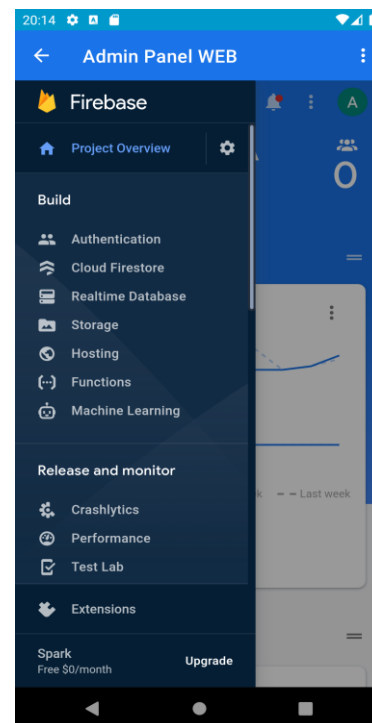


Fig. 4.1. Firebase Admin

4.2. Adăugare utilizatori



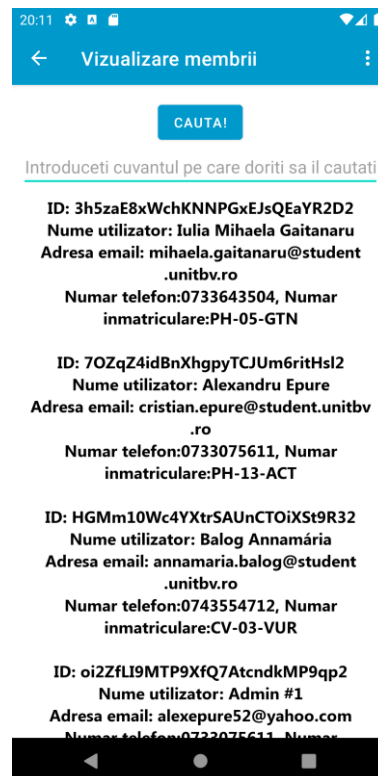
Fig. 4.2.1. Adăugare utilizatori

Oferă posibilitatea adminului de a înregistra noi utilizatori. Trebuie completat:

- Nume;
- E-mail;
- Parola;
- Numărul de telefon;
- Numărul de înmatriculare;
- Să bifeze unul dintre cele 3 căsuțe: elev, profesor, admin.

4.3. Vizualizare membrii

Adminul poate vizualiza și căuta orice utilizator. Lista membrilor este extrasă din baza de date Firebase.



ID	Nume utilizator	Adresa email	Numar telefon	Numar inmatriculare
3h5zaE8xWchKNNPGxEJsQEaYR2D2	Iulia Mihaela Gaitanaru	mihaela.gaitanaru@student.unitbv.ro	0733643504	PH-05-GTN
7OZqZ4idBnXhgyTCJU6ritHsl2	Alexandru Epure	cristian.epure@student.unitbv.ro	0733075611	PH-13-ACT
HGMm10Wc4YXtrSAUnCTOIxSt9R32	Balog Annamária	annamaria.balog@student.unitbv.ro	0743554712	CV-03-VUR
oi2Zfl9MTP9XfQ7AtcndkMP9qp2	Admin #1	alexepure52@yahoo.com	0733075611	

Fig. 4.3.1. Vizualizare membrii

4.4. Modifică membrii

Oferă adminului posibilitatea de a modifica datele personale ale utilizatorilor: adresa de e-mail, numărul de telefon, numărul de înmatriculare, etc.



4.5. Analizare reporturi

În aceasta opțiune se pot vizualiza, genera și șterge reporturi.

Fig. 4.5. Analizare reporturi

4.6. Zona Admin

Adminul poate gestiona zonele și locurile de parcare. Opțiunile acestuia sunt:

- Adăugarea/Ștergerea locurilor de parcare
- Adăugarea/Ștergerea zonelor de parcare

4.7. Contul meu

În această pagină sunt afișate informații generale ale adminului (nr. de telefon, nr. de înmatriculare, istoric parcări, reporturi, parcare activă, mail, nume etc.).

4.8. Resetare parola

Aici adminul își poate reseta parola.

4.9. Butoane adiționale

Interfața de admin oferă funcționalități precum:

- Rezervare parcare
- Vizualizare hartă;

4.10. Algoritm de căutare

```
buttonCauta.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int total = 0;
        String cauta_cuvant =
            editText_cauta.getText().toString().trim().toLowerCase();
        String textComplet = textViewData.getText().toString();
        String[] array = textComplet.split("\n");
        String[] mark = new String[array.length];
        String cuvant;
        StringBuilder st = new StringBuilder();
        for (int i = 0; i < array.length; i++) {
            cuvant = array[i];
            if (cuvant.toLowerCase().contains(cauta_cuvant)) {
                mark[i] = cuvant.trim();
                st.append("<font color='red'><b><i>" + mark[i] +
                    "</i></b></font>");
                total++;
            } else {
                st.append(cuvant);
            }
            st.append("<br>");
        }
        textViewData.setText(Html.fromHtml(" " + st));
        editText_cauta.setText("Aparitii: " + total);
    }
});
```


5. Concluzii

5.1. Concluzii generale

Parcărilor vor rămâne o problemă care afectează întreaga comunitate în ceea ce privește gestionarea traficului. Această aplicație ajută la construirea unui sistem de parcare online. Se concentrează, de asemenea, pe analiza spațiului de parcare și gestionarea controlului traficului. Cu aceasta aplicație, am propus un sistem eficient pentru a efectua parcare online. O parcare bine gestionată poate spori eficiența utilizării parcării.

5.2. Elemente de originalitate

Aplicația a fost dezvoltată și gândită de către membrii echipei, au fost folosite cele mai noi tehnologii atât pe partea de back-end și pe partea de front-end, cât și în privința bazei de date.

O parte pe care o considerăm complexă este preluarea locurilor de parcare din baza de date, existând un adaptor care face acest lucru, întregul proces multiplicându-se în funcție de numărul locurilor de parcare. Statusul fiecărui loc de parcare este actualizat în timp real, ceea ce eficientizează întregul proces de a rezerva o parcare. De asemenea, prin intermediul sistemului de raportare a urgențelor/ inconveniențelor, care operează tot în timp real, oferim utilizatorilor oportunitatea de a primi/ trimite date fără întârziere.

5.3. Direcții viitoare de dezvoltare

Dezvoltarea ulterioară a aplicației „Parcare Colină” va include pe parcurs mai multe funcționalități dezvoltate pentru a sprijini cerințele operatorilor de parcare.

Pe viitor, ne propunem să extindem aria zonelor de parcare, să oferim utilizatorilor opțiunea de chat și, de asemenea, să dezvoltăm un sistem de notificare. Vom include un tutorial video cu scopul de a ajuta utilizatorii să folosească aplicația.

5.4. Impresii

Pe cât de utilă este aplicația în viața șoferilor, pe atât de complexă s-a dovedit a fi realizarea acesteia.

“Acest proiect mi-a dezvoltat abilitățile practice prin interacțiunea cu mediul de dezvoltare oferit de Android Studio și limbajul de programare Java. Am avut un sentiment minunat să rulez pentru prima dată aplicația pe telefon!”

(Iulia)

“Acest proiect m-a ajutat să descopăr complexitatea și frumusețea limbajului de programare Java și, de asemenea, mi-a dezvoltat capacitatea de lucru în echipă.”

(Diana)

“La acest proiect cel mai mult mi-a plăcut dezvoltarea front-endului în Android Studio, lucrul cu baza de date în timp real și faptul că am avut ocazia de a colabora cu colegii mei.”

(Anna)

“Când am început să dezvoltăm acest proiect, ne-am adus aminte de o vorbă: „Suntem atât de mici, într-o lume atât de mare”. Pas cu pas, ne-am adaptat cerințelor, am pus ideile, informațiile și cunoștințele în practică și am reușit să dezvoltăm acest proiect. Interacțiunea cu mediile de dezvoltare Android Studio și Firebase mi-a dezvoltat abilitățile practice și mi-a stârnit interesul de a cunoaște tainele acestor medii și limbajele de programare specifice.”

(Alex)

6. Bibliografia

- [1] ***, <https://firebase.google.com/docs/>
- [2] ***, <https://www.udemy.com/course/invata-limbajul-java-si-android-studio/>
- [3] ***, <https://developer.android.com/docs>
- [4] ***, <https://developer.android.com/guide/topics/ui/layout/recyclerview>
- [5] ***, <https://cloud.google.com/firestore/docs>
- [6] ***, <https://firebase.google.com/docs/database>