

# DOCUMENTAȚIA PROIECTULUI “LOST PETS”

**Studenti:**

Gramadă Ioana Cătălina – 10LF292

Andronache Victor - 10LF291

## 1. Prezentarea proiectului

Ți-ai pierdut animalul de companie și nu știi cum să îl găsești mai repede? Proiectul nostru se numește “Lost Pets” și reprezintă o aplicație prin intermediul căreia utilizatorii pot posta și vizualiza anunțuri cu animale pierdute.

Aplicația se deschide cu o pagina de unde se poate alege între a te autentifica cu un cont deja existent, a-ți crea un cont, sau a vedea poze random cu animale de companie. După ce utilizatorul se înregistrează cu un email și o parolă, acesta se poate autentifica pentru a vedea și adăuga anunțuri. După logare, pot fi îndeplinite următoarele acțiuni: adăugare anunț, vizualizarea tuturor anunțurilor, vizualizarea anunțurilor proprii și deconectare.

La adăugarea unui anunț trebuie specificat numele animalului, rasa acestuia și locul în care a fost pierdut. De asemenea, este necesară și încărcarea unei poze cu animalul respectiv, pentru a fi mai ușor de identificat. O dată cu adăugarea anunțului, se introduce automat și adresa de email cu care sunteți autentificat, pentru a putea fi contactat în cazul în care animalul este găsit.

Dacă alegeți să vizualizați toate anunțurile, vă va apărea o listă cu toate animalele pierdute înregistrate în aplicație, împreună cu adresele de email ale proprietarilor, iar dacă doriți să vizualizați numai anunțurile dumneavoastră, veți avea și posibilitatea de a le șterge, în cazul în care animalul a fost găsit.

Atunci când ți-ai găsit animalul, poți reveni pe prima pagină pentru a te relaxa vizualizând câteva poze random cu animale de companie.

## 2. Tehnologii folosite

Am folosit **Firestore Authentication** pentru înregistrarea utilizatorilor şi logarea acestora în cont. Firestore Authentication facilitează construirea de sisteme de autentificare sigure, îmbunătăţind în acelaşi timp experienţa de conectare şi integrare pentru utilizatorii finali. Oferă o soluţie de identitate end-to-end, acceptând conturi de e-mail şi parolă. Construită de aceeaşi echipă care a dezvoltat Google Sign-in, Smart Lock şi Chrome Password Manager, securitatea Firestore aplică expertiza internă Google a gestionării uneia dintre cele mai mari baze de date de cont din lume.

Pentru a stoca anunţurile cu animale pierdute, am creat o **baza de date locală cu Room**. Biblioteca de persistenţă oferă un strat de abstractizare asupra SQLite pentru a permite accesul fluent la baza de date, în timp ce pune în valoare întreaga putere a SQLite. În special, Room oferă verificarea în timp de compilare a interogărilor SQL.

Am utilizat **Android Location Service** pentru a prelua locaţia curentă, precompletând câmpul cu locaţia atunci când se adaugă un anunţ. Android oferă această funcţionalitate utilizând serviciile de localizare (location service). Acestea oferă acces la facilităţi ce pot fi folosite pentru a determina poziţia curentă cunoscute sub numele de location providers.

### 3. Request-uri folosite

Am realizat **Http Requests** folosind **Volley**. Aceasta este o bibliotecă HTTP care face reţeaua pentru aplicaţii Android mai uşoară şi, mai rapidă. Am preluat datele sub formă de obiecte de tip **JSON Object**, pe care le-am prelucrat şi le-am încărcat într-un RecyclerView. Obiectele JSON sunt construite în perechi de tip cheie: valoare. Elementele obiectului sunt separate prin virgule, fiecare obiect este delimitat cu ajutorul acoladelor, iar un şir de obiecte este memorat între paranteze pătrate. Este un standard de scriere mai economic decât XML în ceea ce priveşte mărimea datelor, dar mai puţin lizibil decât acesta, limitat la valori text şi valori numerice, nefiind posibilă memorarea informaţiilor în format binar.

### 4. Arhitectura

Aplicaţia conţine 3 activităţi şi 7 fragmente. Prima activitate, “WelcomeActivity” cuprinde fragmentele de Login şi Register. Acestea folosesc Firebase Authentication. În fragmentul Register se introduc datele din EditText-urile corespunzătoare email-ului şi parolei abia după ce au fost efectuate validări asupra lor. În fragmentul Login se încearcă autentificarea la Firebase, returnând mesaj de succes, sau de eroare dacă credenţialele nu au fost introduse corect.

Următoarea activitate, “Second Activity” conţine un fragment “FragmentMain” cu butoane şi fragmentele ce se deschid în urma apăsării lor: fragmentele de adăugare şi vizualizare a anunţurilor. Acestea folosesc o baza de date locală creată cu Room. Baza de date are o entitate numită “AnnouncementItem” ce conţine datele despre animal: numele acestuia, rasa, zona în care a fost văzut ultima dată, email-ul proprietarului, uri-ul pozei cu acesta şi un id folosit ca şi cheie primară.

Inserarea, ştergerea şi extragerea anunţurilor din baza de date le-am descris în AnnouncementDAO, acesta reprezentând un mijloc de comunicare dintre baza de date şi aplicaţie, prin intermediul entităţilor. Pentru realizarea acestor metode am folosit Async Task-uri şi un Repository cu ajutorul căruia am accesat AnnouncementDatabase şi AnnouncementDAO.

Fragmentul prin care se face inserarea în baza de date oferă utilizatorului posibilitatea de a alege o poză din galeria sa, memorand-ui Uri-ul. De asemenea, câmpul rezervat locaţiei este precompletat cu locaţia curentă a user-ului.

Pentru afişarea anunţurilor în RecyclerView am folosit două adaptoare. Pentru fragmentul FragmentView am folosit adaptorul AllAnnouncementsAdapter cu ajutorul căruia am pus informaţiile din baza de date într-un RecyclerView. În celălalt fragment, FragmentViewMine, am preluat din baza de date doar anunţurile postate de userul conectat în acel moment (anunţurile pentru care adresa de email a proprietarului animalului este aceeaşi ca a userului conectat), iar ca şi adaptor am folosit AnnouncementAdapter, acesta având în plus faţa de AllAnnouncementsAdapter un buton de “Delete” prin apăsarea căruia se şterge anunţul respectiv din baza de date.

Fragmentul “FragmentMain” conţine, pe lângă butoanele de adăugare şi vizualizare, şi un buton de “Sign Out” prin apăsarea căruia user-ul se deconectează şi revine la prima activitate.

Am mai creat o activitate, “RandomPictureActivity” prin intermediul căreia am afişat într-un RecyclerView obiecte luate de la adresa “<https://loremflickr.com/json/320/240/pets>” realizând HttpRequests. Obiectele sunt in format JSON, conţinând url-ul unei poze cu un animal şi numele proprietarului. Pentru acestea am folosit un adapter, “HttpRequestAdapter”.

## 5. Utilizarea aplicatiei

Aplicaţia se deschide cu o fereastră ce conţine 3 butoane: Log In, Register si View Random Pets. Utilizatorii fără cont, pot apăsa pe ultimul buton, vazând imagini random cu animale însoţite de numele proprietarului. Daca utilizatorul doreşte să vadă sau să adauge anunţuri cu animale pierdute, acesta se poate inregistra apăsând butonul de Register. Acesta va fi redirecţionat catre o fereastră unde îşi va introduce email-ul si o parola cu care se va autentifica in aplicatie. Butonul de Log In trimite utilizatorul catre o fereastră unde îşi introduce credentialele.

După autentificare, acesta poate:

- Să introducă anunturi – va introduce numele animalului pierdut şi rasa acestuia. Aplicaţia oferă posibilitatea de a adăuga locaţia curentă ca şi locaţia în care a fost văzut ultima dată animalul. În cazul în care utilizatorul doreşte să introducă altă locaţie, va modifica în caseta respectivă.

De asemenea, se va introduce şi o poză cu animalul pierdut din galeria personală pentru a fi mai uşor de recunoscut. După introducerea acestor date, apăsând pe butonul de adaugare, se va adauga anuntul, putând fi vazut atunci când apasă pe butoanele de vizionare.

- Să vizualizeze anunţuri – se va deschide o fereastră cu toate anunţurile cu animale pierdute din aplicaţie, însoţite de adresele de email ale proprietarilor pentru a-i putea contacta daca a fost văzut animalul.
- Să-si vizualizeze propriile anunţuri - se va deschide o fereastră doar cu anunţurile postate de el pe care le poate şterge
- Să se deconecteze – urmând a fi direcţionat la fereastra cu care s-a deschis aplicaţia

## 6. Concluzii

Proiectul de faţă are ca scop prezentarea unei aplicaţii care să ajute persoanele care şi-au pierdut animalele de companie să le recupereze - avându-se în vedere facilităţile puse la dispoziţie de sistemul de operare Android.

Din punct de vedere structural, o aplicaţie Android este compusă din trei nivele principale: partea de interfaţă pe care o persoană o poate vedea şi cu care poate interacţiona, partea logică care obţine informaţiile cerute de utilizator şi partea de date cu ajutorul căreia sunt salvate informaţiile necesare logicii aplicaţiei.

Rolul acestui program este de a încuraja oamenii să nu renunțe la animalele lor după ce au fost pierdute, fapt ce ar rezulta creșterea la număr a animalelor strazii. Aplicația realizează o comunicare mai ușoară între oameni, aceștia ajutându-se între ei pentru a-și găsi animalele de companie. Pentru acestea, au fost utilizate serviciile de localizare puse la dispoziție de Android, platforma Firebase și baza de date Room.

În concluzie, consider că platforma Android pune la dispoziție instrumentele necesare pentru crearea unor aplicații care să ajute la o mai bună interrelaționare a oamenilor, oferind elementele necesare pentru realizarea unui interfețe care să atragă utilizatorii și cu care aceștia să poată interacționa ușor.

**Link GitHub:** <https://github.com/Catalina1808/LostPets>

**Link Trello:** <https://trello.com/b/XsBh3znm/lostpet>