

RAPORT DE DESCRIERE ARHITECTURALĂ - COLECTAREA CERINȚELOR -

Popescu Paullo Robertto Karloss 406 IS

Cuprins

1. Scopul aplicației	3
2. Aria de acoperire a aplicației	3
3. Grupurile de interese	4
4. Colectarea și analiza cerințelor	4
5. Prioritizarea cerințelor	5
6. Specificații de analiză arhitecturală	6
7. Tactici arhitecturale.....	6
8. Specificații de proiectare	7
9. Alegerea tehnologiilor	8
10. Perspective arhitecturale.....	8
a. Perspectiva de utilizare.....	8
b. Perspectiva asupra datelor	16
c. Perspectiva structurală.....	18
d. Perspectiva comportamentală.....	18
e. Perspectiva fizică.....	20

1. Scopul aplicației

Descriere:

RescuePets este o aplicație mobilă Android destinată facilitării procesului de adopție a animalelor. Scopul principal este de a crește numărul de adopții și de a simplifica întregul proces, ajutând adăposturile să gestioneze mai eficient cererile și interacțiunea cu potențialii adoptatori.

Misiune:

- Să aducă o soluție digitală complementară procesului tradițional de adopție, punând accent pe comunicarea directă între utilizatori și angajații adăposturilor.
- Să ofere un mediu sigur și ușor de utilizat pentru toți cei interesați, indiferent de nivelul de cunoștințe tehnice.

Utilizatori:

- Persoane interesate de adopția animalelor.
- Adăposturi și centre de protecție a animalelor, precum și angajații acestora.
- Personalul de suport tehnic responsabil cu întreținerea aplicației.

2. Aria de acoperire a aplicației

RescuePets este o interfață digitală ce pune în legătură potențialii adoptatori cu adăposturile de animale, oferind profiluri detaliate ale animalelor (vârstă, rasă, istoric, poze) și un chat cu asistenți virtuali (angajați). Nu este un serviciu de vânzare online și nici nu înlocuiește complet vizita la centru.

Obiectivele principale vizează:

- Comunicare în timp real prin chat cu personalul adăpostului, integrată în aplicație. Asistenții (angajații adăpostului) pot răspunde la întrebări, oferi sfaturi și îndruma utilizatorii. Această funcționalitate asigură consiliere și suport pe parcursul procesului de adopție și are și un rol educativ;
- Profiluri detaliate ale animalelor disponibile pentru adopție, cu informații esențiale și fotografii, astfel încât utilizatorii să poată cunoaște bine fiecare animal înainte de a decide;
- Programarea vizitelor online pentru a vedea animalul dorit (aceste cereri sunt aprobate sau respinse de către personal în funcție de disponibilitate și de anumite criterii) și pentru evitarea aglomerației;
- Metode avansate de căutare/filtrare a animalelor, pentru ca viitorii adoptatori să găsească rapid un companion potrivit preferințelor lor (specie, rasă, vârstă, talie etc.);
- Facilitarea unei adopții responsabile (informații despre nevoi/îngrijire).

Prin acestea, *RescuePets* eficientizează procesul, crescând șansele ca fiecare animal să-și găsească un cămin potrivit.

3. Grupurile de interese

Stakeholders principali:

- **Utilizatorii finali:** Persoanele care doresc să adopte animale și care folosesc aplicația pentru a găsi și programa vizite la adăposturi.
- **Adăposturile/Centrele de protecție a animalelor:** Instituții ce gestionează adopțiile și care beneficiază de un sistem care le optimizează fluxul de lucru.
- **Angajații adăposturilor:** Personalul care aprobă programările și interacționează direct cu utilizatorii prin chat și alte funcționalități.
- **Dezvoltatorii și echipa tehnică:** Responsabili cu mentenanța, actualizările și asigurarea securității aplicației.
- **Investitori și parteneri:** Organizații non-profit sau sponsori interesați de promovarea adopțiilor și de soluții tehnologice inovatoare în domeniul protecției animalelor.

4. Colectarea și analiza cerințelor

Metodologia de colectare:

- **Experiența personală și cunoștințele directe:** Analiza problemelor actuale din procesul de adopție și identificarea nevoilor specifice atât ale utilizatorilor, cât și ale centrelor de adopție.
- **Documentare online:** Studiarea unor aplicații similare din domeniu și analiza tendințelor pieței (ex. statisticile privind căutările online pentru "pet adoption").
- **Analiza aplicațiilor similare:** Evaluarea funcționalităților și a feedback-ului de la utilizatori pentru a identifica avantajele și neajunsurile soluțiilor existente.

Cerințe funcționale:

- **Autentificare și înregistrare:** Permite utilizatorilor crearea de conturi și accesul securizat la aplicație.
- **Profilul animalelor:** Afișarea detaliată a informațiilor (specie, rasă, an de naștere, descriere, imagini) pentru fiecare animal disponibil pentru adopție.
- **Căutare și filtrare:** Funcționalitate de căutare avansată pentru a găsi rapid animalul potrivit în funcție de diverse criterii (specie, locație, etc.).
- **Programarea vizitelor:** Modul prin care utilizatorii pot trimite cereri de vizită, care ulterior vor fi aprobate sau respinse de către angajați.
- **Chat interactiv:** Comunicare în timp real între utilizatori și asistenții din cadrul adăposturilor pentru clarificarea detaliilor legate de adopție.
- **Sincronizarea datelor:** Implementarea unui sistem hibrid, ce asigură funcționarea offline (prin baza de date Room) și online (prin Firebase).

Cerințe de calitate:

- **Securitate:**

- Stocarea parolelor prin metode de hashing pentru protejarea datelor sensibile.
- Restricționarea accesului la date, astfel încât utilizatorii să poată accesa doar propriile informații.

- **Performanță și disponibilitate:**

- Utilizarea tehnicilor de caching (Room) pentru a asigura funcționarea aplicației chiar și în lipsa conexiunii la internet.
- Sincronizarea eficientă între baza de date locală și cea online, pentru a evita blocajele sau erorile de comunicare.

- **Ușurință în utilizare (UX/UI):**

- Implementarea unui design modern și intuitiv, folosind principii de Material Design și proiectarea realizată în Figma.
- Navigație simplă și feedback vizual clar pentru utilizator.

Surse de colectare:

- Documentația tehnică și de implementare din aplicația *RescuePets*.
- Analize de piață și studii de caz ale aplicațiilor de adopție existente.
- Experiența personală în dezvoltarea de aplicații mobile Android, inclusiv deciziile arhitecturale (MVVM, Clean Architecture) și tehnologiile utilizate (Room, Firebase, Java).

5. Prioritizarea cerințelor

Din lista de cerințelor de calitate, consider că următoarele trei sunt cele mai relevante pentru succesul aplicației *RescuePets*:

- **Securitatea datelor:**

Justificare: Având în vedere natura sensibilă a informațiilor personale (datele utilizatorilor, detalii ale animalelor, istoricul programărilor), este esențial ca aplicația să asigure protecția maximă a datelor. Implementarea unor măsuri de securitate solide (ex. stocarea parolelor cu hashing, restricții de acces) va construi încrederea utilizatorilor și va proteja integritatea sistemului.

- **Performanța și disponibilitatea offline:**

Justificare: Pentru a oferi o experiență de utilizare continuă, aplicația folosește o bază de date locală (Room) pentru caching și sincronizarea cu Firebase. Astfel, utilizatorii vor putea accesa informațiile stocate chiar și în lipsa conexiunii la internet. Totuși, experiența offline nu va oferi toate funcționalitățile disponibile online (cum ar fi actualizările în timp real, chat-ul interactiv sau alte operațiuni dinamice), dar asigură accesul la datele esențiale și o utilizare de bază în

condiții de conectivitate redusă.

- **Ușurința în utilizare (UX/UI):**

Justificare: O interfață intuitivă și un design modern sunt critice pentru adoptarea pe scară largă a aplicației. Un proces de navigație simplu, combinat cu elemente vizuale atractive (realizate cu Material Design și proiectarea în Figma), va facilita interacțiunea utilizatorilor și va crește rata de adopție, contribuind la succesul global al aplicației.

6. Specificații de analiză arhitecturală

Scenarii de calitate:

- *Securitatea datelor:* „Ca utilizator, vreau ca parola mea să fie stocată securizat folosind metode avansate de hashing, pentru a preveni accesul neautorizat și compromiterea datelor personale.”

- Ca utilizator, vreau ca accesul meu să fie restricționat pe bază de roluri clare, pentru a asigura confidențialitatea datelor sensibile și protecția împotriva abuzurilor.
- Deducere: din misiunea de a oferi un mediu sigur și protejat pentru utilizatori și animale.

- *Performanță și disponibilitate offline:* „Ca utilizator, vreau să pot accesa informațiile esențiale despre animale și cererile mele chiar și fără conexiune la internet, pentru a avea continuitate în procesul de adopție.”

- Ca utilizator, vreau sincronizarea automată a datelor între baza locală și online, pentru a evita inconsistența datelor și a avea acces constant la informații actualizate.
- Deducere: din obiectivul aplicației de a facilita un proces continuu și fără întreruperi pentru utilizatori.

- *Ușurința în utilizare (UX/UI):* „Ca utilizator nou, vreau ca aplicația să aibă o interfață intuitivă și ușor navigabilă, astfel încât să pot găsi rapid informațiile dorite fără ajutor suplimentar.”

- Ca utilizator, vreau o interfață clară și vizual atractivă pentru a îmbunătăți experiența mea generală și a face procesul de adopție mai plăcut și eficient.
- Deducere: din obiectivul aplicației de a atrage un număr cât mai mare de adoptatori printr-o experiență plăcută și accesibilă.

7. Tactici arhitecturale

Securitatea datelor:

- Utilizarea algoritmului bcrypt pentru hashing-ul parolelor.
- Implementarea sistemului de autentificare bazat pe roluri. Justificare: bcrypt asigură protecție robustă împotriva atacurilor de tip brute-force, iar autentificarea bazată pe roluri permite gestionarea eficientă și securizată a permisiunilor și accesului la informații.pP

Performanță și disponibilitate offline:

- Folosirea bazei de date locale Room pentru stocarea și accesarea datelor esențiale offline.
- Integrarea cu Firebase pentru sincronizarea automată și transparentă între bazele de date. Justificare: Room asigură acces rapid și performanță ridicată în mod offline, iar Firebase oferă sincronizare continuă și actualizări eficiente ale datelor în timp real.

Ușurința în utilizare (UX/UI):

- Adoptarea principiilor Material Design pentru o interfață coerentă și atractivă.
- Realizarea testelor de utilizabilitate și ajustarea interfeței pe baza feedback-ului utilizatorilor. Justificare: Material Design asigură o interfață intuitivă și atractivă, facilitând adoptarea rapidă a aplicației, iar feedback-ul periodic permite îmbunătățiri constante.

8. Specificații de proiectare

Securitatea datelor:

- „Ca dezvoltator, vreau să implementez hashing bcrypt pentru parolele utilizatorilor, astfel încât parolele să nu fie niciodată stocate în clar și să asigur un nivel ridicat de securitate împotriva atacurilor.”
- „Ca dezvoltator, vreau să implementez un sistem bazat pe roluri clare (utilizator simplu, angajat), pentru ca fiecare utilizator să poată accesa exclusiv informațiile și funcționalitățile permise rolului său.”

Performanță și disponibilitate offline:

- „Ca dezvoltator, vreau să folosesc baza de date locală Room pentru stocarea datelor esențiale (profiluri animale, programări, informații personale), pentru a permite accesul rapid și neîntrerupt al utilizatorilor chiar și atunci când conexiunea la internet lipsește.”
- „Ca dezvoltator, vreau să implementez sincronizarea automată între Room și Firebase, astfel încât atunci când utilizatorul revine online, datele să se actualizeze imediat și automat, evitând inconsistențele.”

Ușurința în utilizare (UX/UI):

- „Ca designer, vreau să creez o interfață conform principiilor Material Design și să realizez teste iterative de utilizabilitate cu utilizatorii finali, astfel încât aplicația să fie intuitivă și ușor de utilizat încă de la prima interacțiune.”
- „Ca dezvoltator, vreau să implementez feedback vizual (animații discrete, mesaje clare de eroare și confirmare) și o structură clară a navigației, pentru a reduce timpul de adaptare și a îmbunătăți experiența generală de utilizare.”

9. Alegerea tehnologiilor

- **Android Studio (Java)**

Justificare: Android Studio oferă suport extins și optimizat pentru dezvoltarea aplicațiilor native Android, asigurând performanță și stabilitate.

- **Firebase (Autentificare, Realtime Database, Cloud Storage)**

Justificare: Firebase permite sincronizarea eficientă a datelor și gestionarea securizată a autentificării.

- **Room Database**

Justificare: Room permite caching eficient al datelor, contribuind la performanța aplicației în modul offline.

- **Retrofit**

Justificare: Retrofit simplifică integrarea API-urilor și gestionarea sincronizării datelor între serviciile online și locale.

- **Material Design Framework**

Justificare: Material Design asigură o interfață coerentă și prietenoasă, îmbunătățind experiența utilizatorilor.

- **Figma**

Justificare: Figma permite prototiparea și colaborarea eficientă pentru designul interfeței.

10. Perspective arhitecturale

a. Perspectiva de utilizare

Perspectiva de utilizare evidențiază interacțiunile esențiale ale utilizatorilor cu aplicația, facilitând o navigare intuitivă și clară:

- **Diagrame Wireframe** realizate în **Figma**, ilustrând interfețele principale ale aplicației:
 - Ecran autentificare și înregistrare
 - Pagina principală (vizualizare animale disponibile)
 - Pagina de profil a animalelor
 - Pagina de chat interactiv
 - Pagina pentru programarea vizitelor

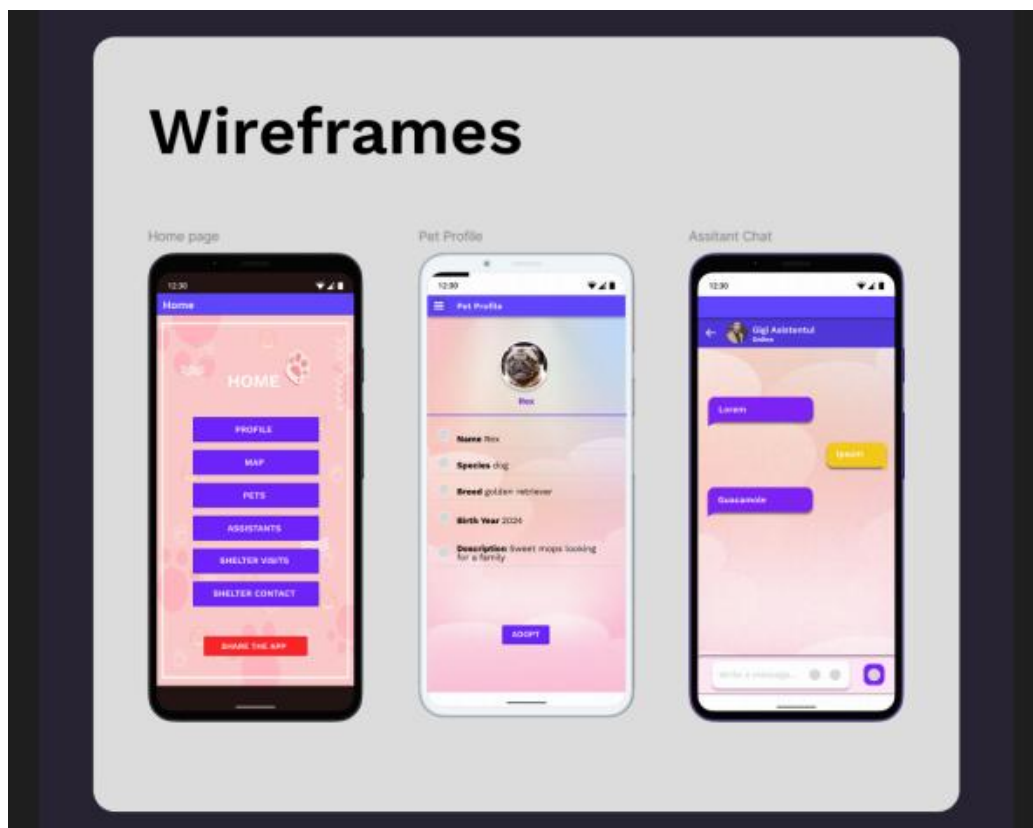


Fig. 1. Prototip realizat în Figma.

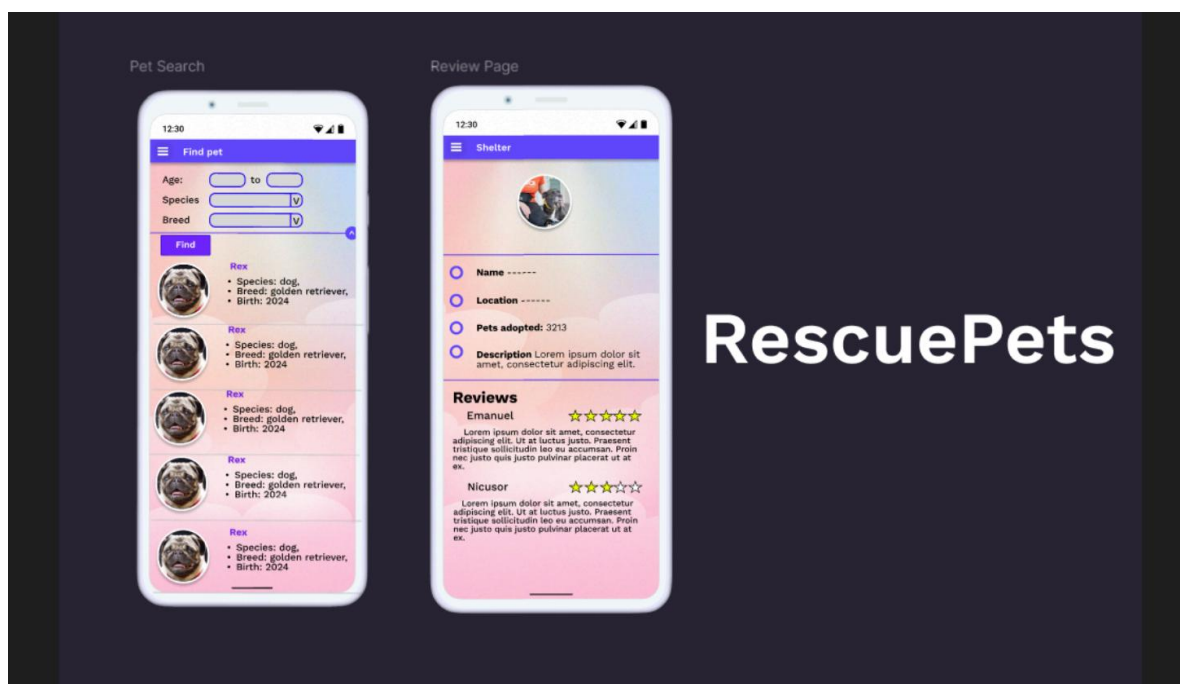


Fig. 2. Prototip realizat în Figma.

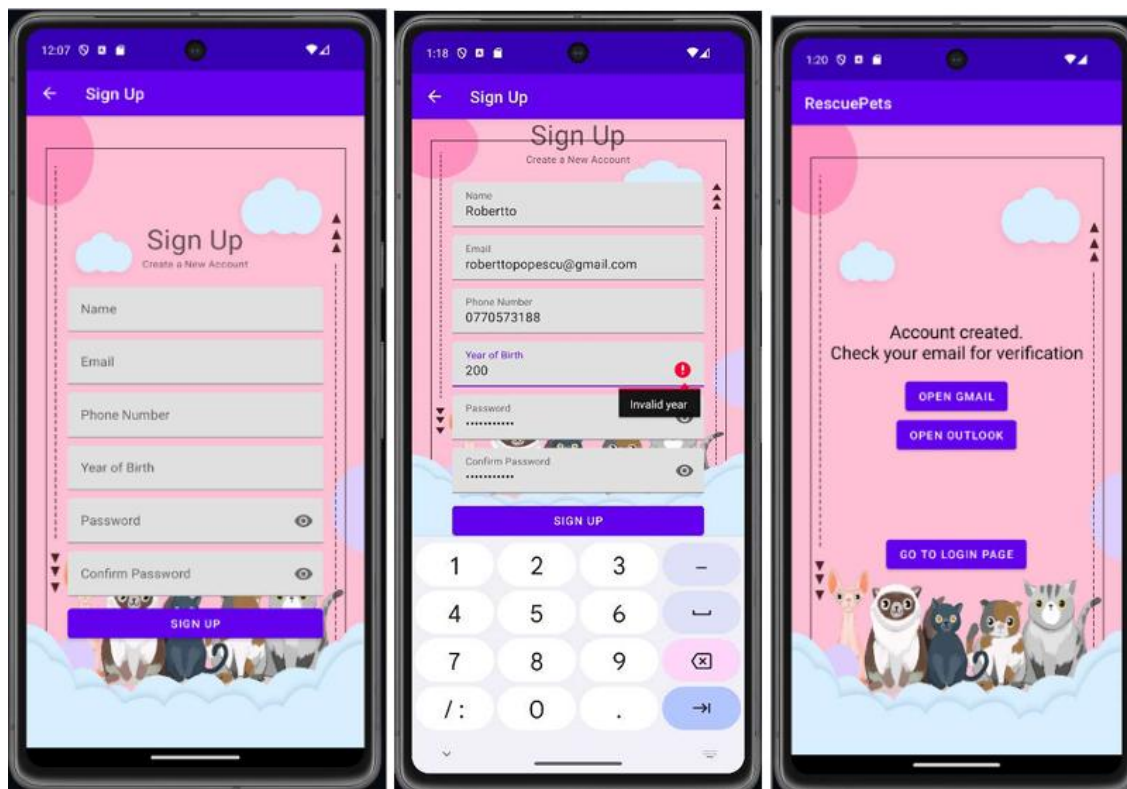


Fig. 3. Pagina de înregistrare, validare an naștere, confirmare înregistrare.

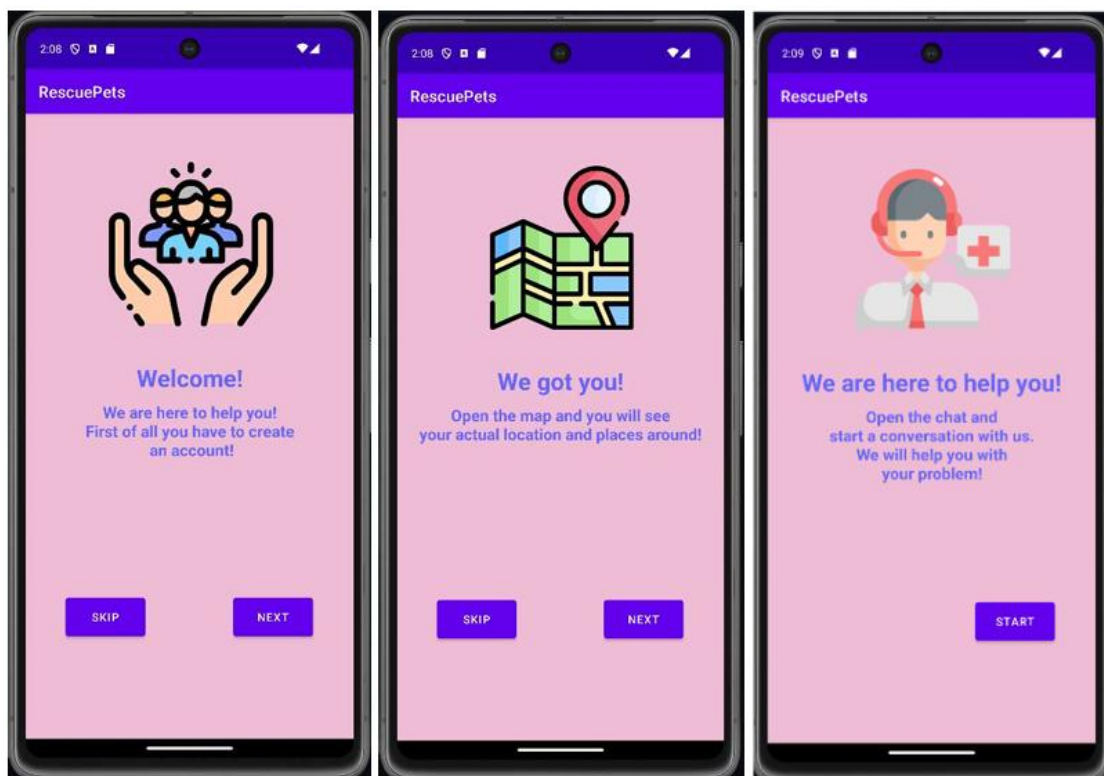


Fig. 4. Ghidul de introducere al aplicației.

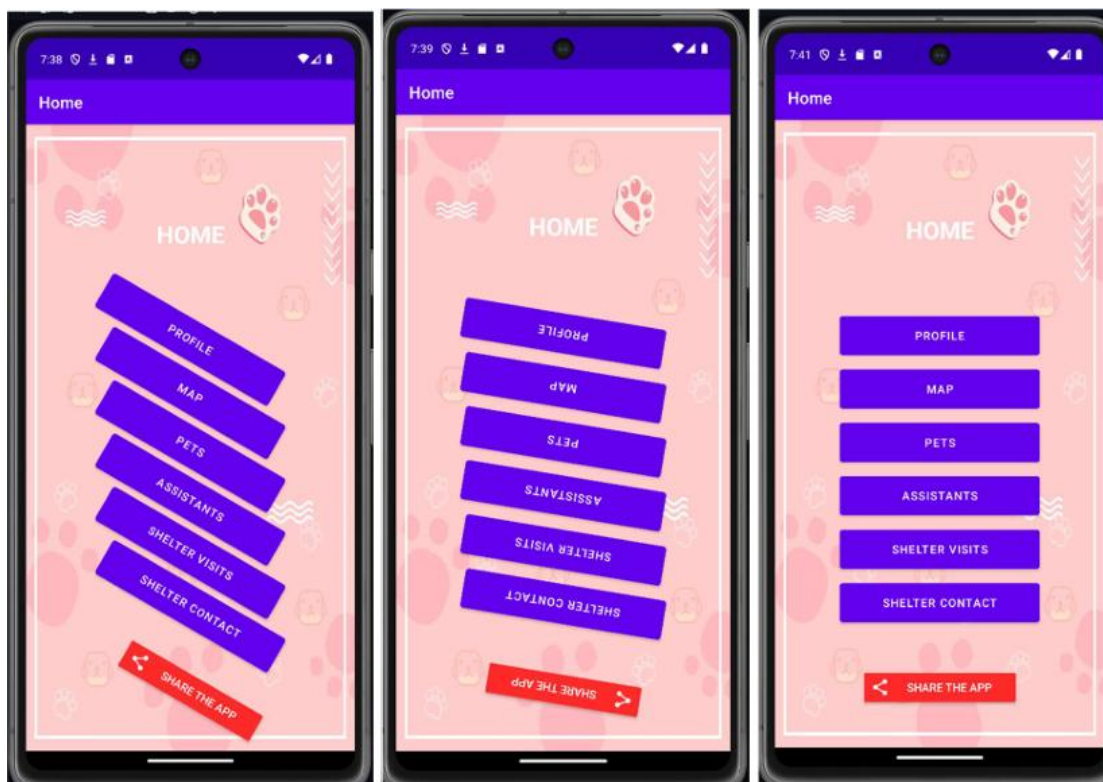


Fig. 5. Pagina principală a aplicației, unde a fost aplicat ObjectAnimator pentru animația pe butoane.

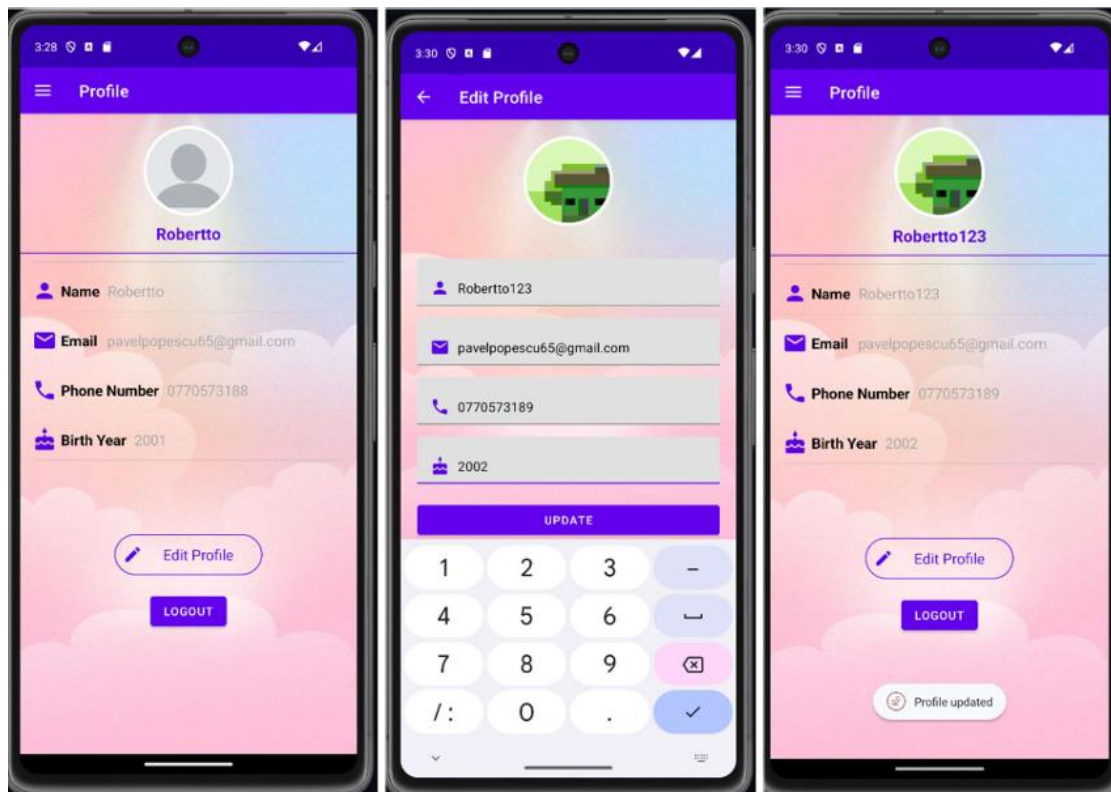


Fig. 6. Pagina de profil și editare profil a utilizatorului curent.

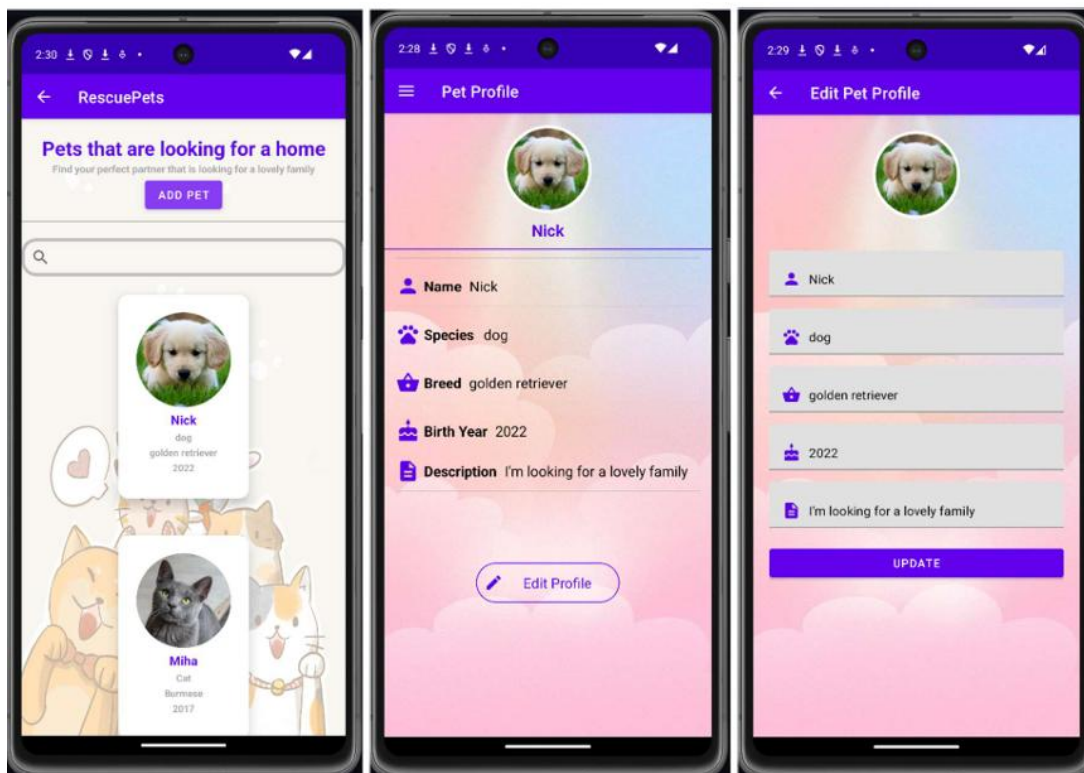


Fig. 7. Funcționalitate angajaților de adăugare/editare a animalelor în centru.

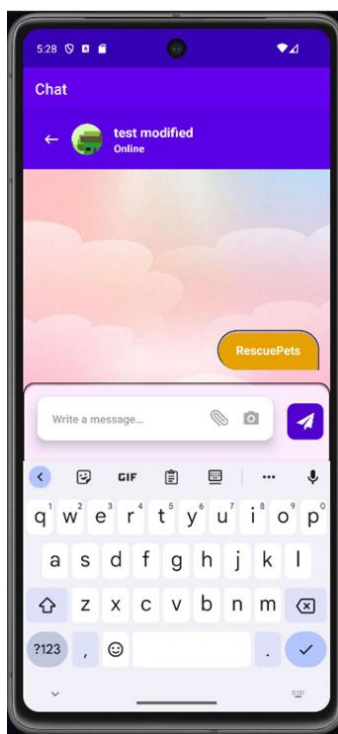


Fig. 7. Pagina de chat.

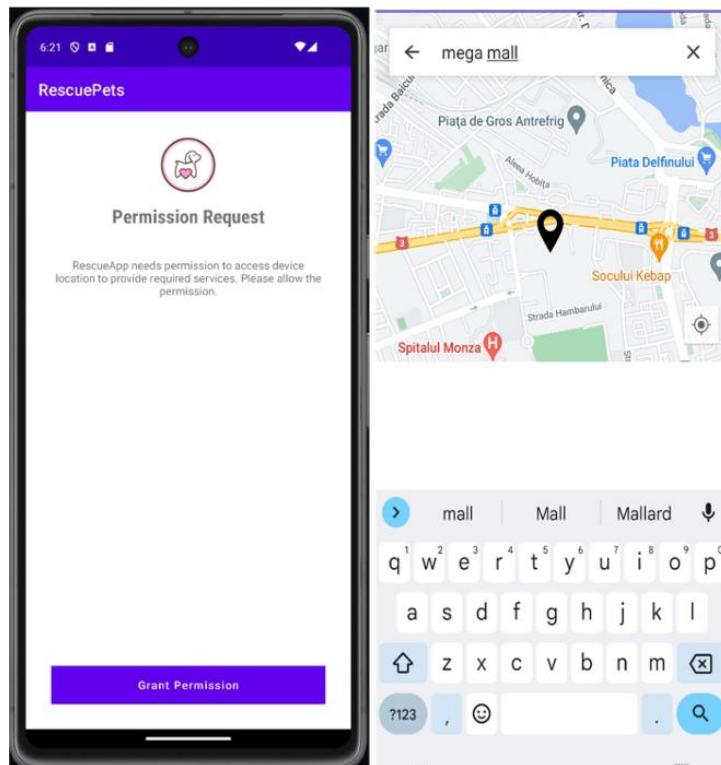


Fig. 8. Harta.

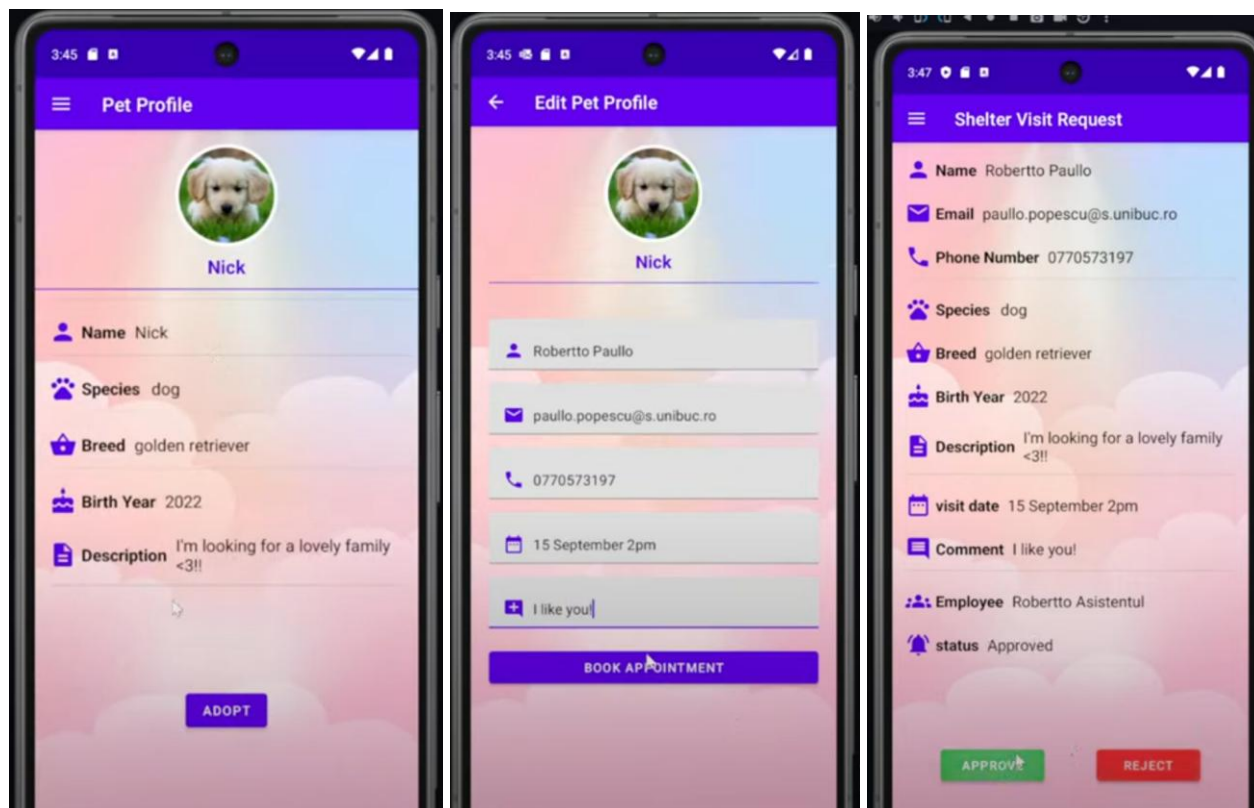


Fig. 9. Pagina de programare vizită animal în centru.

- **Diagrame UML de stări** pentru fluxurile cheie:

- Procesul de autentificare și înregistrare
- Procesul de căutare, filtrare și navigare animale
- Fluxul programării unei vizite și gestionarea statusului cererilor (în așteptare, aprobat, respins)

Aceste elemente asigură o experiență prietenoasă și eficientă pentru utilizatori, contribuind la cerința de calitate „Ușurință în utilizare”.

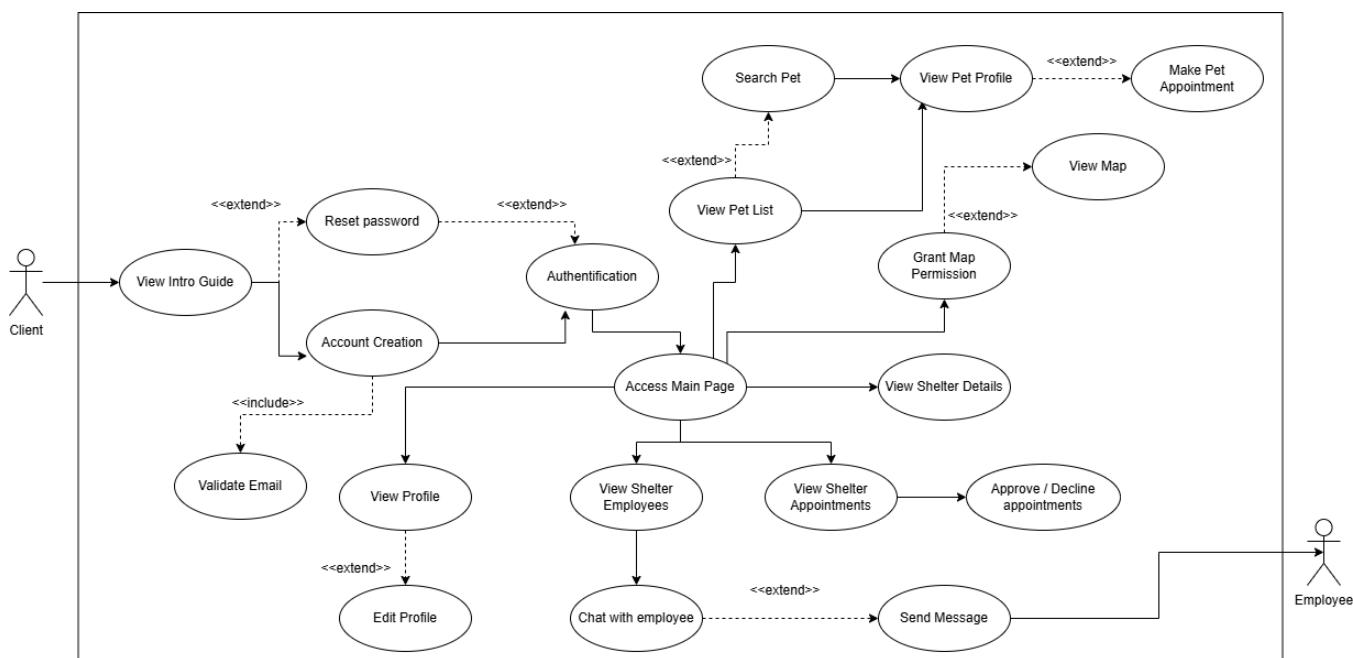


Fig. 10. Diagrama Use Case.

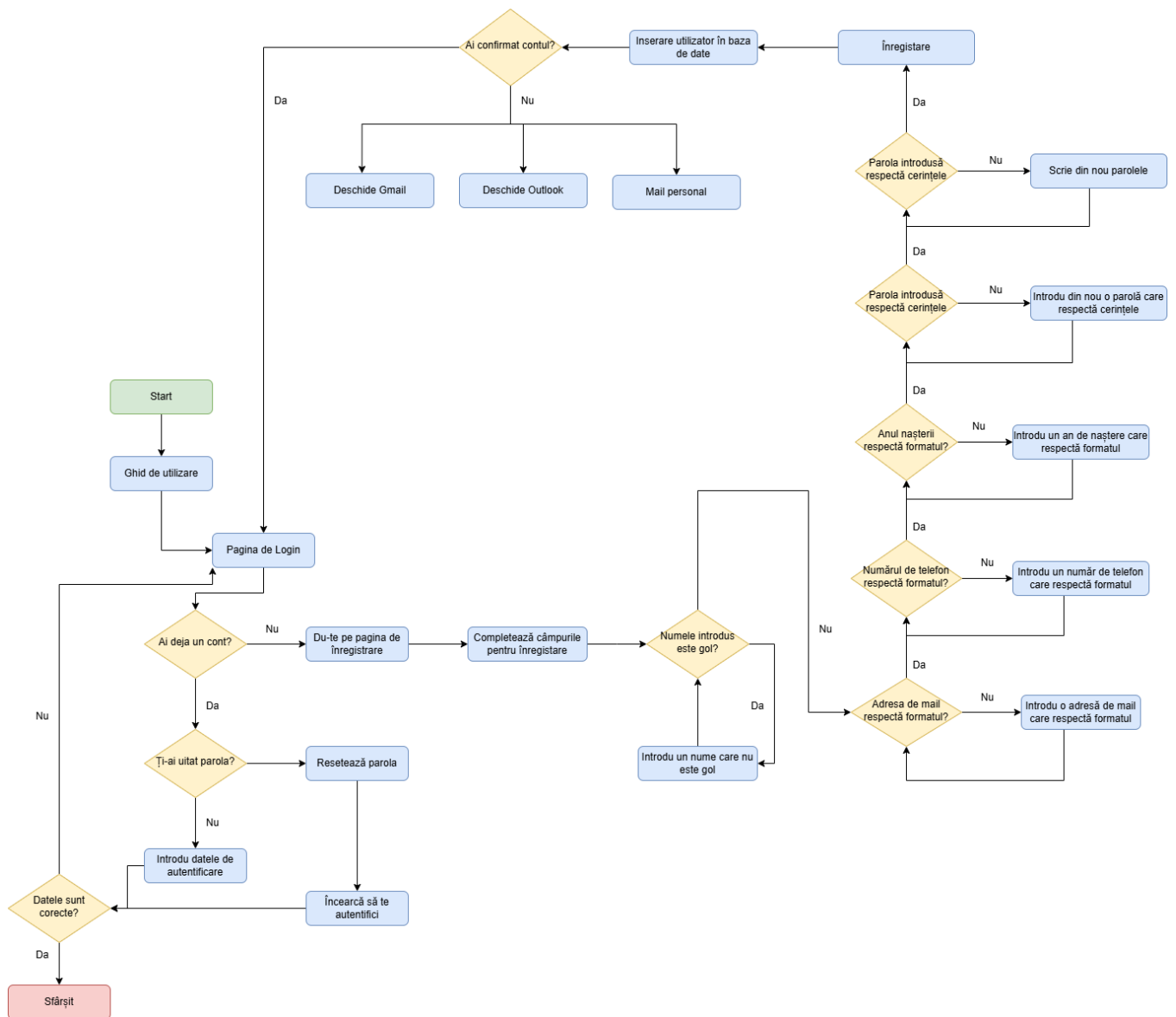


Fig. 11. Diagrama de flux pentru procesul de autentificare, înregistrare, resetare parolă.

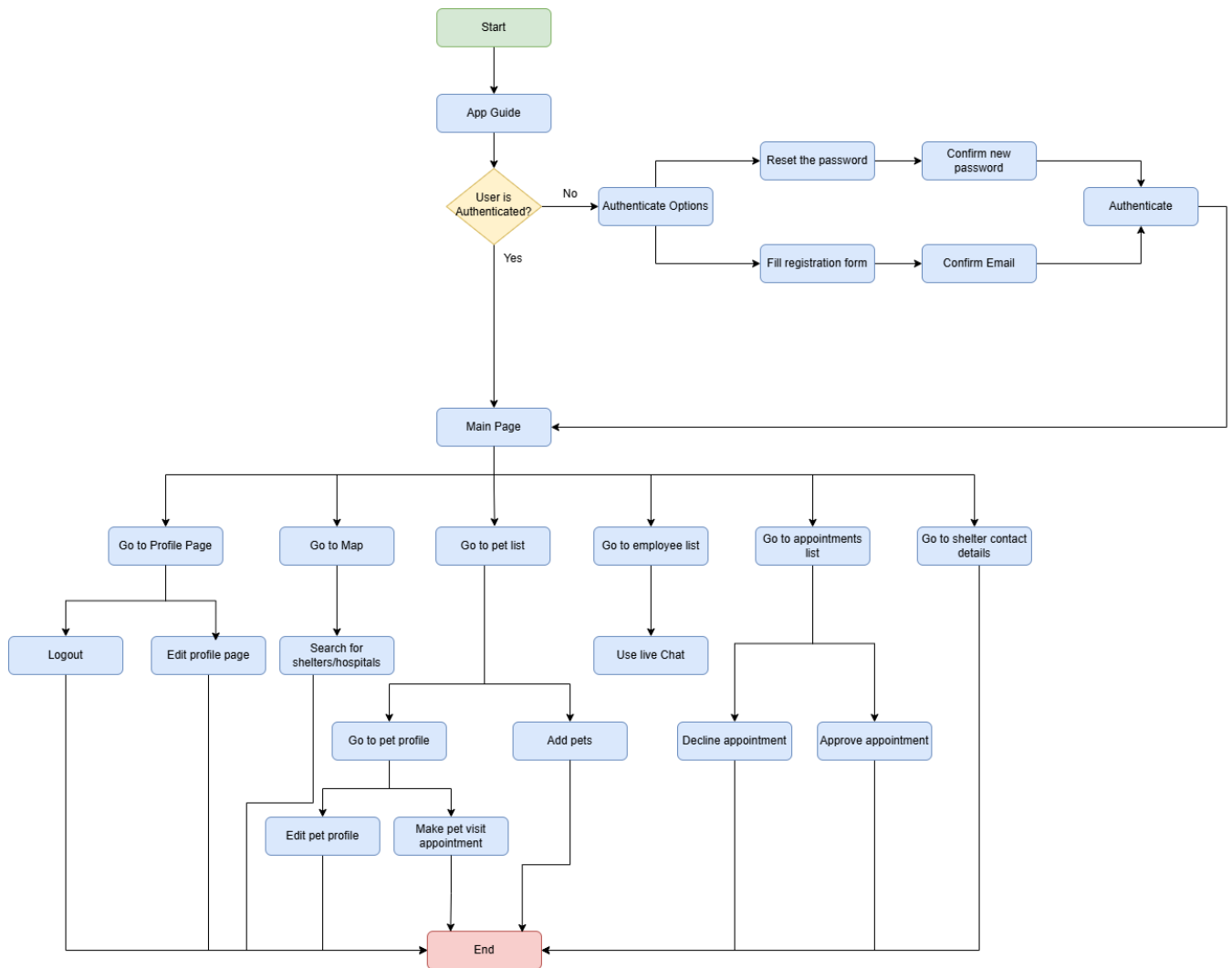


Fig 12. Diagrama de flux general al aplicației.

b. Perspectiva asupra datelor

Perspectiva asupra datelor descrie structura logică și organizarea datelor în cadrul aplicației RescuePets:

- **Diagramă conceptuală UML (ERD)** care conține următoarele entități principale și relațiile dintre ele:
 - **Utilizator** (id, nume, email, parolă, telefon, imagine, rol)
 - **Angajat** (id, nume, rol, centru_adăpost_id)
 - **Animal** (id, nume, specie, rasă, vârstă, descriere, imagini, centru_adăpost_id)
 - **Centru Adăpost** (id, nume, adresă, telefon, email, program)
 - **Cererile de vizită** (id, data_programată, status, utilizator_id, animal_id, angajat_id)

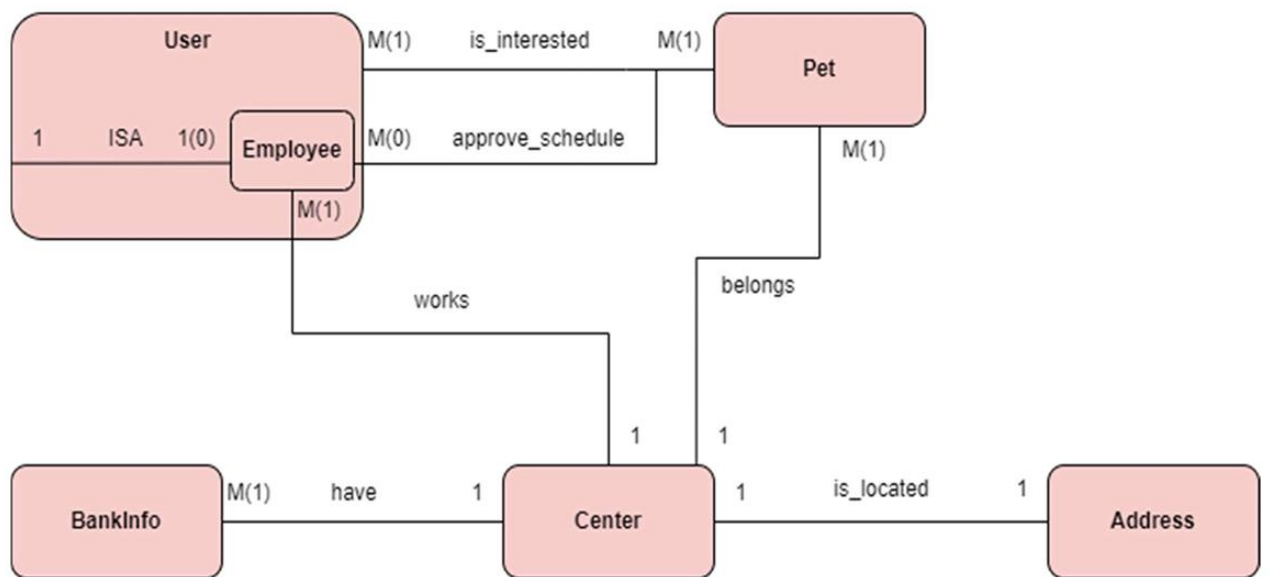


Fig. 13. Diagrama ERD.

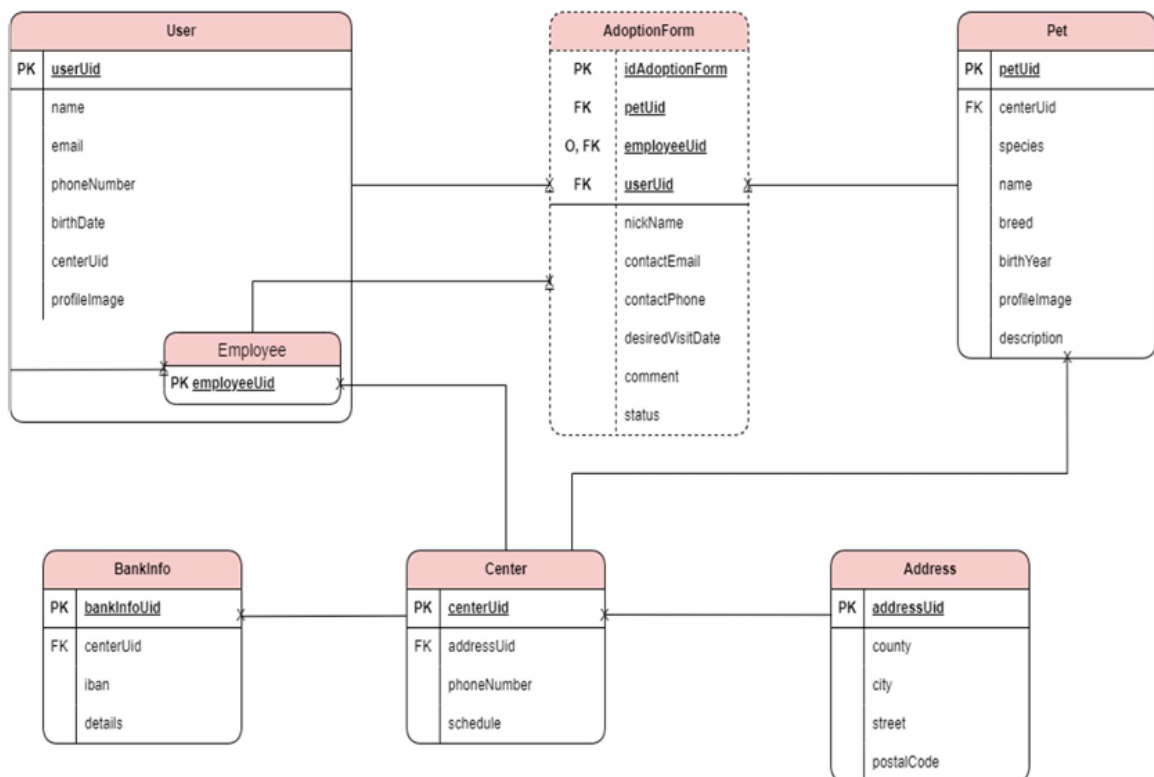


Fig. 14. Diagrama Conceptuală.

Datele sunt gestionate atât local (**Room Database**), cât și online (**Firebase Realtime Database**), asigurând performanța și disponibilitatea offline.

Această perspectivă asigură o structură logică eficientă și bine definită, acoperind tacticile privind performanța și disponibilitatea datelor.

c. Perspectiva structurală

Perspectiva structurală descrie componentele software și modul în care acestea interacționează, conform modelului arhitectural adoptat (MVVM și Clean Architecture):

- **View** (interfața grafică):
 - Responsabilă cu afișarea informațiilor și capturarea interacțiunilor utilizatorilor.
 - Interacționează cu ViewModel.
- **ViewModel**:
 - Gestionează logica aplicației și menține starea datelor între View și Model.
 - Utilizează LiveData pentru a actualiza automat interfața când datele se modifică.
- **Repository** (pattern repository):
 - Intermediază accesul la sursele de date (Firebase, Room).
 - Centralizează operațiunile CRUD și sincronizarea între bazele de date.
- **Data Source (Local și Remote)**:
 - LocalDataSource (Room) pentru stocare locală.
 - RemoteDataSource (Firebase) pentru stocare și sincronizare online.
- **Mediator**:
 - Gestionează sincronizarea automată între LocalDataSource și RemoteDataSource.
- **Dependency Injection (Dagger)**:
 - Asigură injectarea automată și sigură a dependențelor între componente.

Această structură clară și modulară contribuie la securitatea datelor, performanța și mentenabilitatea aplicației.

d. Perspectiva comportamentală

Perspectiva comportamentală detaliază procesele interne ale aplicației:

- **Diagrame de secvență și activitate UML** pentru procese critice:
 - Procesul de autentificare:

- Introducerea datelor → validarea în Firebase → răspunsul sistemului.
- Procesul de sincronizare date offline-online:
 - Actualizarea locală → detectarea modificărilor → sincronizarea cu Firebase.
- Procesul de programare a vizitelor:
 - Trimiterea cererii de către utilizator → notificarea angajatului → aprobarea/respingerea → notificarea utilizatorului.

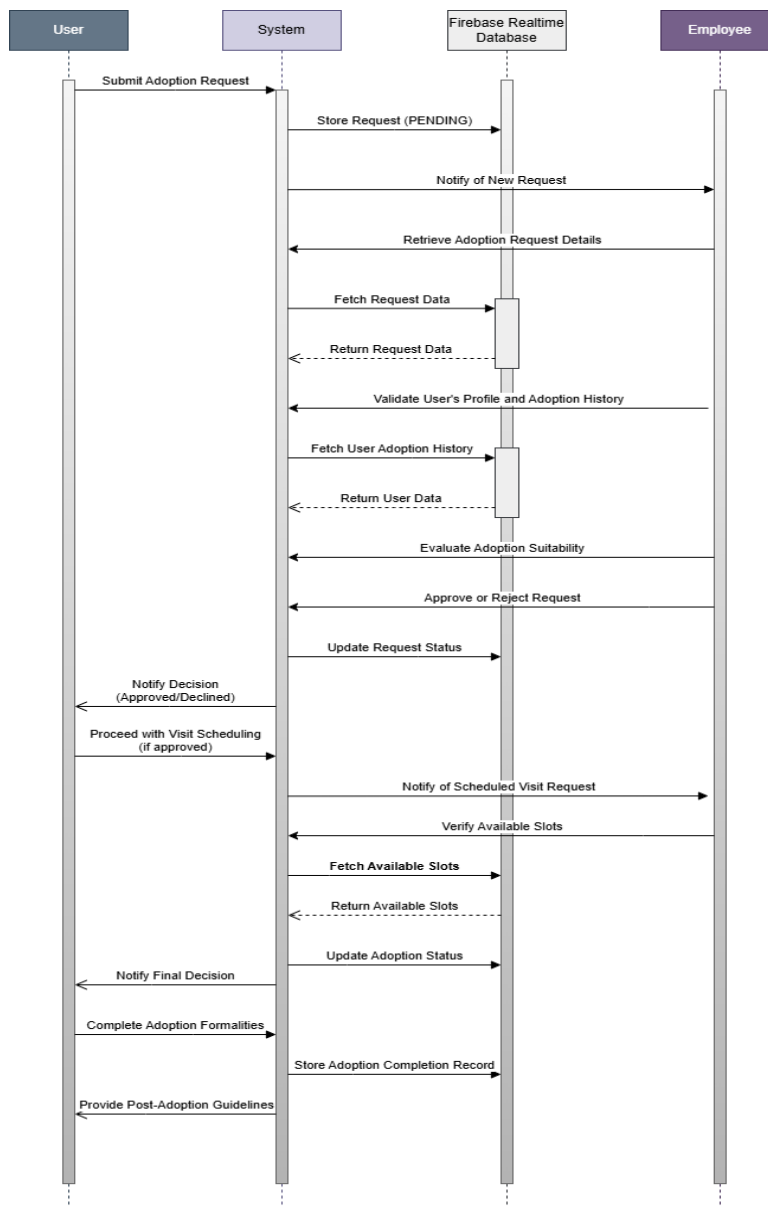


Fig. 15. Diagrama de secvență

Aceste procese comportamentale clar definite asigură o gestionare eficientă a fluxului de lucru

intern, acoperind tacticile pentru performanță și securitate.

e. Perspectiva fizică

Perspectiva fizică descrie mediul hardware și software în care va rula aplicația:

- **Diagrama de Deployment UML:**
 - **Client mobil Android** (dispozitive Android API 26+)
 - Include aplicația RescuePets, Room Database pentru stocare offline.
 - **Firestore Cloud** (server online)
 - Realtime Database pentru sincronizare și stocare date online.
 - Firebase Authentication pentru gestionarea autentificării securizate.
 - Firebase Cloud Storage pentru gestionarea imaginilor și documentelor atașate.
 - **Comunicarea:**
 - Protocoale securizate HTTPS și API REST prin Retrofit.

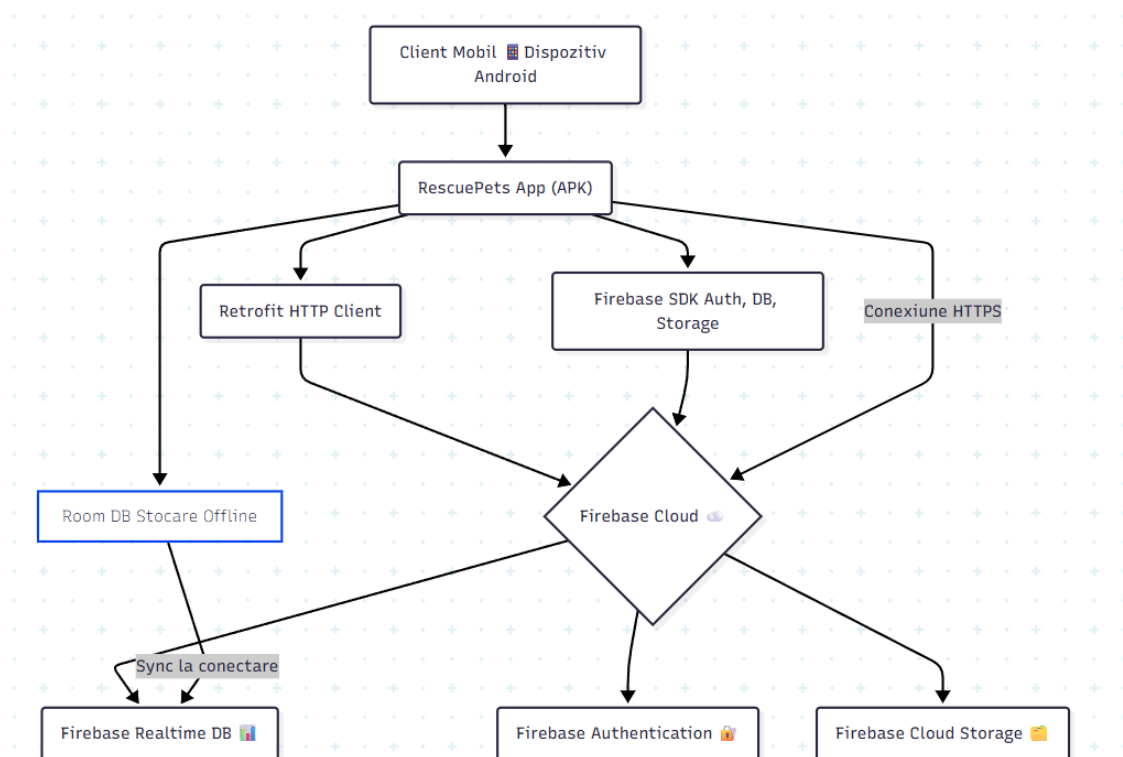


Fig. 16. Diagrama de deployment

Această configurație oferă claritate în gestionarea componentelor tehnice, contribuind la performanța, securitatea și disponibilitatea continuă a aplicației.