



RescuePets



Student:
Popescu Paullo-Robertto Karloss
406 IS



Cuprins



01 Introducere

02 Tehnologii folosite

03 Cerinte functionale si non functionale



04 Diagrame

05 Design patterns

06 Demo-ul aplicatiei





01

Introducere



400 milioane

Animale care nu au un stăpân





3 milioane

Anual sunt eutanasiate





02 Tehnologii folosite



Tehnologii folosite



Android



Firebase



Room



Java



Gradle cu Kotlin DSL





03 Cerinte functionale





| Cerința | Justificare |
|--|---|
| Trebuie să mă pot autentifica și să îmi pot reseta parola | Pentru a-mi menține securizat contul și a putea recupera accesul dacă am uitat parola. |
| Trebuie să pot autentifica/inregistra cu contul de Google | Pentru a facilita accesul rapid și convenabil, folosind serviciile Google. |
| Trebuie să mă pot deloga de pe cont | Pentru a asigura securitatea contului meu. |
| Trebuie să pot vizualiza și actualiza profilul meu (nume, email, telefon, poză etc.) | Pentru a menține informațiile mele personale corecte și actualizate. |
| Trebuie să pot căuta și vizualiza detalii despre animalele disponibile (specie, rasă, vârstă, descriere, imagini) | Pentru a găsi mai ușor animalul potrivit nevoilor și preferințelor mele. |
| Trebuie să pot trimite cereri de programare la centrul de animale (pentru vizită/adopecție) | Pentru a stabili data și ora vizitei în adăpost și a facilita procesul de adopție. |
| Trebuie să pot vedea statusul cererilor de programare (în așteptare, aprobat, respins) | Pentru a fi la curent cu evoluția și rezultatul solicitărilor mele. |
| Trebuie să pot comunica în timp real cu un angajat (chat/asistent virtual) | Pentru a primi informații și sfaturi rapide despre procesul de adopție sau îngrijirea animalelor. |
| Trebuie să pot trimite atașamente (poze, documente) în chat către un angajat | Pentru a facilita comunicarea și a oferi documente relevante în procesul de adopție. |
| Trebuie să pot șterge mesaje din chat (only for me, delete for everyone) | Pentru a gestiona istoricul conversațiilor în funcție de preferințele mele. |
| Trebuie să pot primi notificări despre evenimente majore (primire mesaj chat, adopție, aprobare cerere, respingere cerere) | Pentru a fi informat în timp util și a nu pierde informații importante. |





| Cerința | Justificare |
|---|--|
| Trebuie să pot accesa harta și să îmi văd locația în timp real | Pentru a găsi mai ușor centrele de adopție și a mă orienta. |
| Trebuie să pot posta anunțuri (în cazul în care doresc să dau spre adopție un animal) | Pentru a facilita găsirea unui nou stăpân și a oferi vizibilitate anunțurilor mele. |
| Trebuie să pot vizualiza și filtra animalele după numele acestora, rasă, specie, anul nașterii | Pentru a găsi cât mai rapid ce animale satisfac dorințele mele. |
| Trebuie să pot încărca și actualiza imagini cu animalele (doar pentru angajați) | Pentru a menține informațiile din platformă la zi și a prezenta animalele cât mai atractiv. |
| Trebuie să pot aproba sau respinge cererile de programare (doar pentru angajați) | Pentru a organiza vizitele și a valida cererile de adopție în mod controlat. |
| Trebuie să pot vedea istoricul cererilor trimise (utilizator) sau istoricul cererilor primite (angajat) | Pentru a urmări evoluția adopțiilor și a ține evidența programărilor. |
| Trebuie să pot vedea informațiile de contact și programul centrului de animale | Pentru a putea contacta centrul și a ști când este disponibil. |
| Trebuie să pot dona sau sponsoriza un centru de animale | Pentru a susține financiar adăpostul și a contribui la îngrijirea animalelor fără stăpân. |
| Trebuie să am un tab cu tutorial video al aplicației | Pentru a învăța mai ușor cum se folosește aplicația și a vedea funcționalitățile în acțiune. |
| Trebuie să pot da share aplicației către prietenii mei | Pentru ca aceștia să o poată găsi mai ușor și să o instaleze. |





03 Cerinte non-functionale





Interfață și UX

- Aplicația trebuie să fie intuitivă și ușor de utilizat, cu o organizare clară a elementelor și aliniată la principiile Material Design.

Performanță

- Paginile și funcționalitățile principale (ex.: căutare animale, vizualizare profil) trebuie să se încarce în câteva secunde pentru a nu afecta experiența utilizatorilor.

Scalabilitate

- Arhitectura (MVVM + Clean Architecture) trebuie să permită extinderea ușoară a funcționalităților și gestionarea unui număr mare de utilizatori/conexiuni simultane.

Securitate

- Parolele utilizatorilor trebuie stocate prin algoritmi de hash, iar accesul la date și funcționalități trebuie să fie controlat prin roluri (utilizator simplu/angajat).



Disponibilitate și Reziliență

- Aplicația trebuie să funcționeze și în modul offline, folosind o bază de date locală (Room) care să mențină o copie a datelor din Firebase.

Compatibilitate

- Aplicația trebuie să ruleze pe dispozitive cu sistem de operare Android, începând de la API 26 (Android 8.1) și până la versiunile recente (Android 14).



Mentenabilitate

- Codul trebuie să fie organizat pe principii solide (MVVM, repository pattern, dependency injection etc.) pentru a facilita întreținerea și actualizările ulterioare.



04 Diagrame

Diagrama Use-Case (generală)

Diagrame de flux

Diagramele de baze de date

Diagrama de clase

Diagrama de obiecte

Diagrama de secvență





Diagrama Use-Case

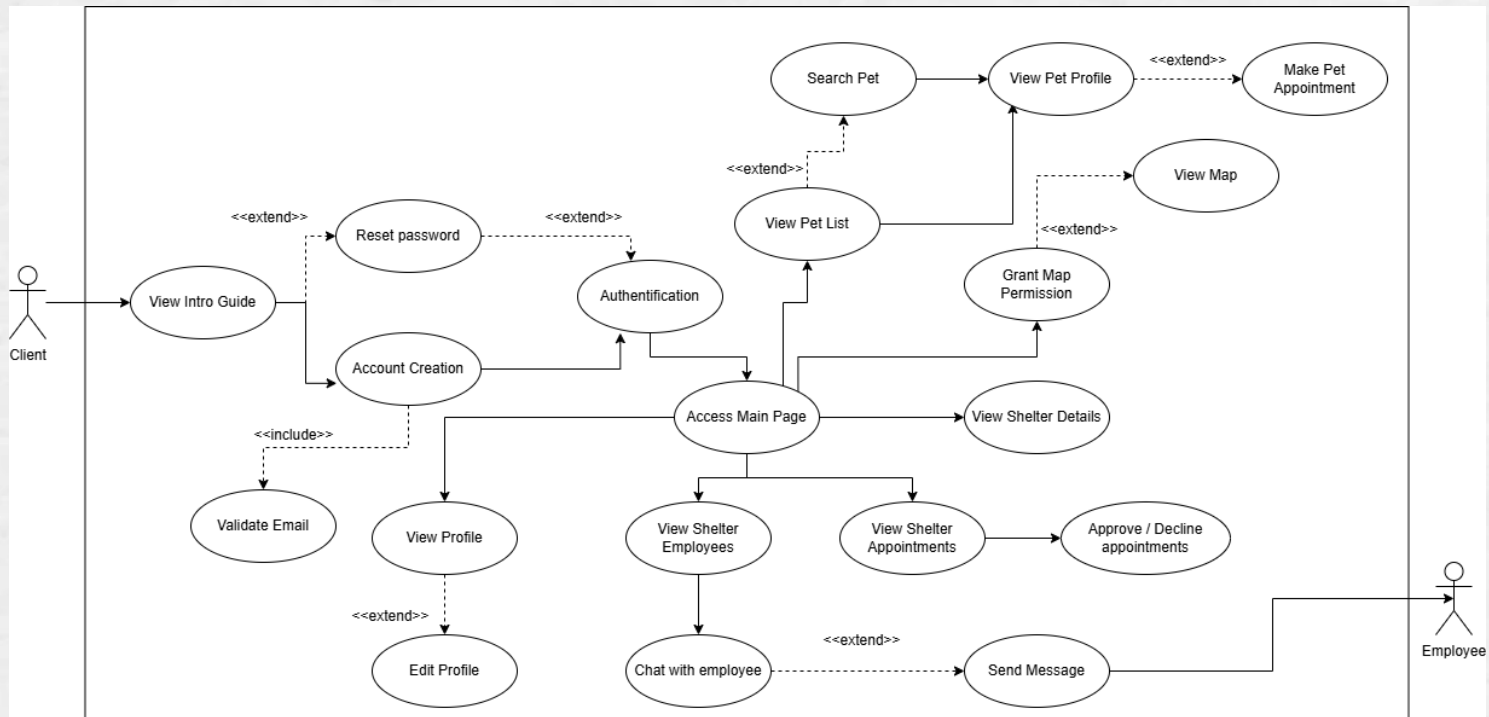




Diagrama Entitate-Relatie- ERD

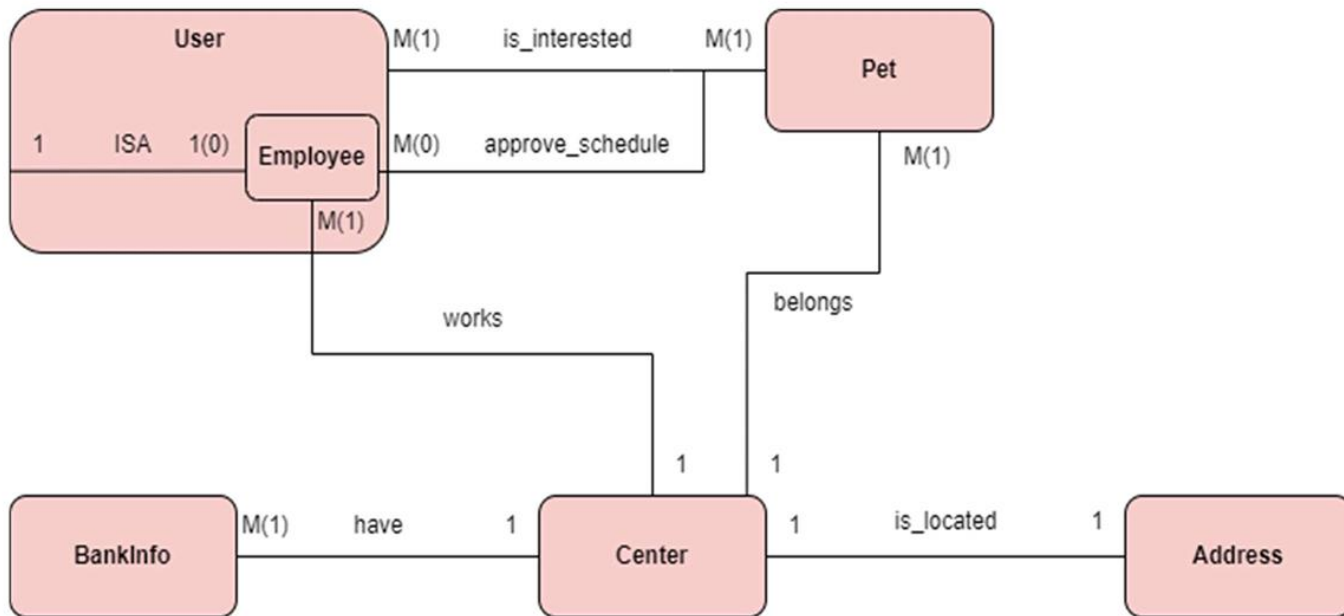




Diagrama Conceptuala

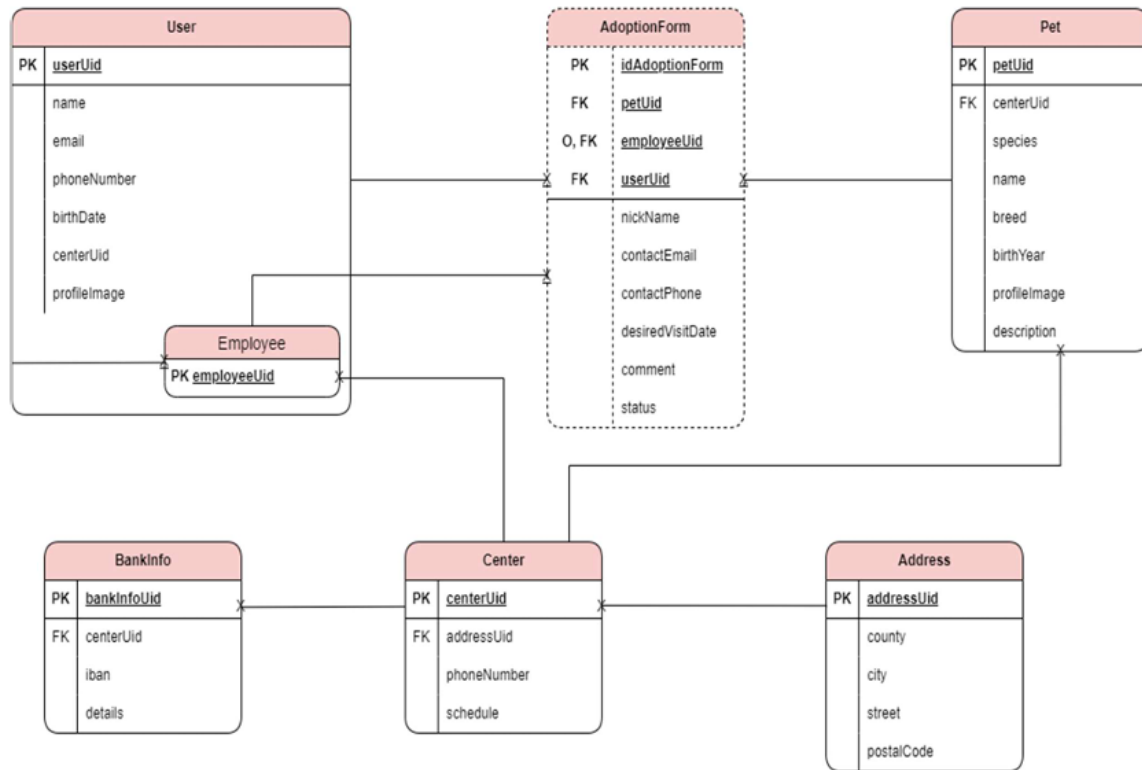




Diagrama de flux

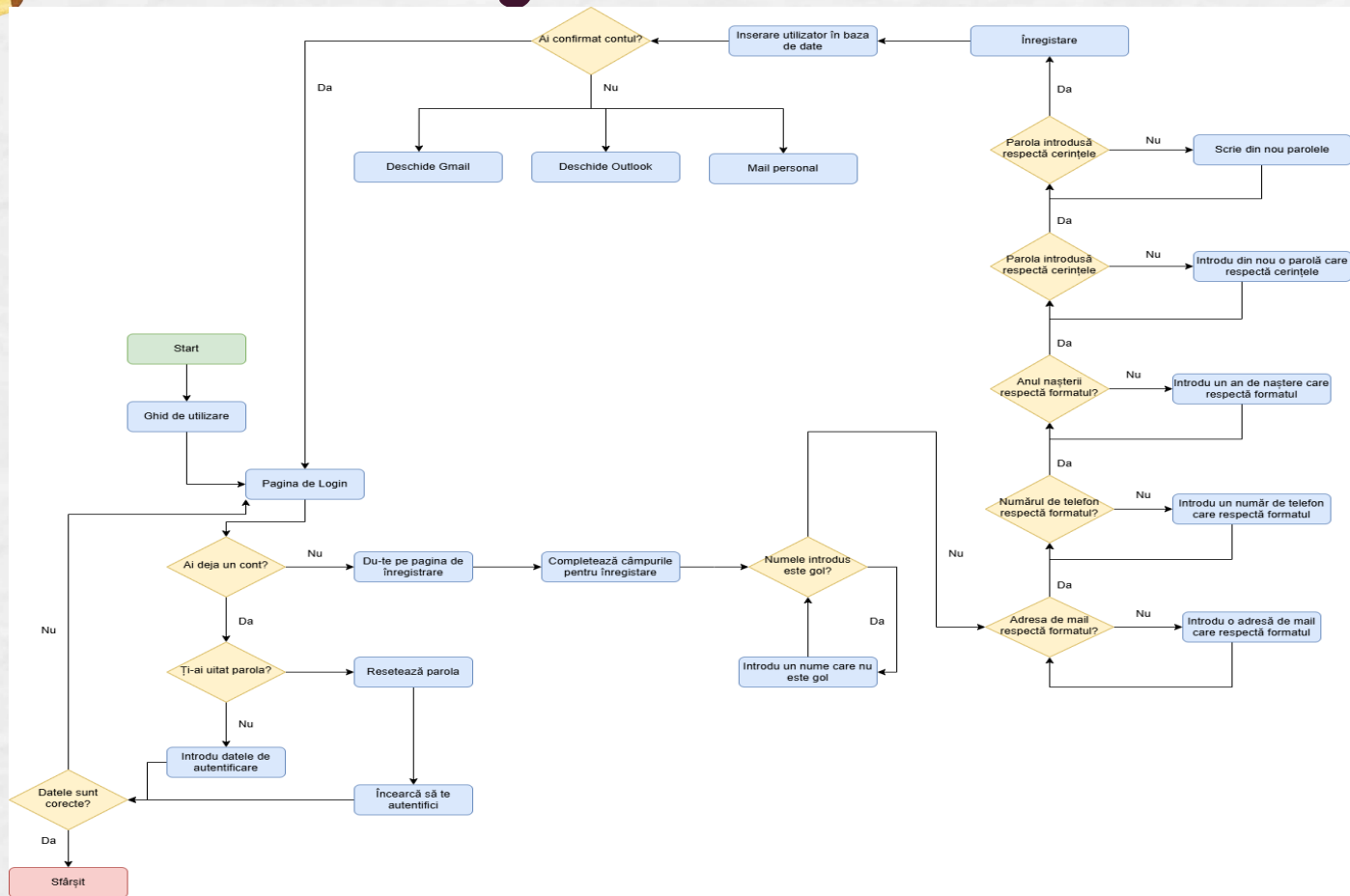




Diagrama de flux

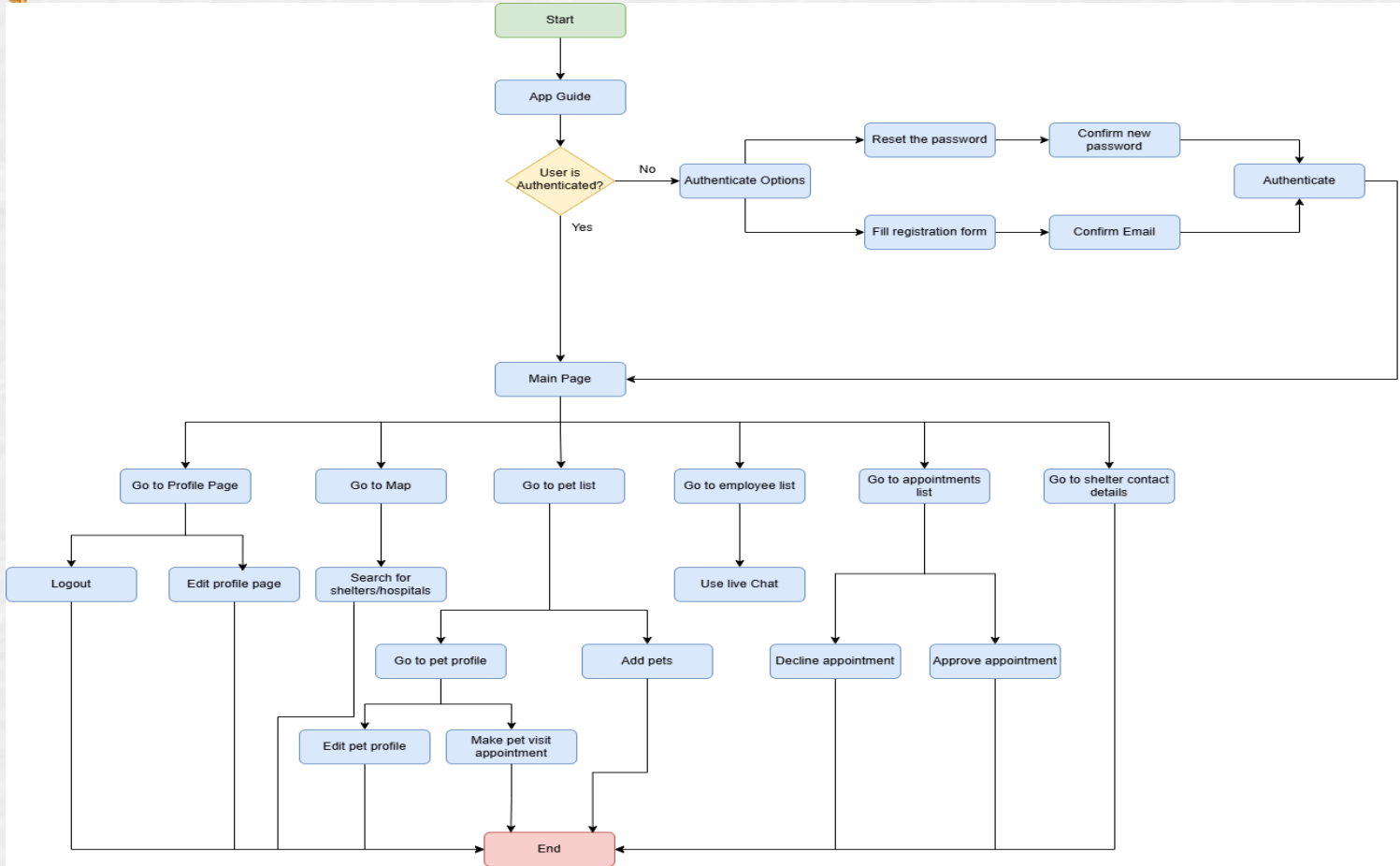
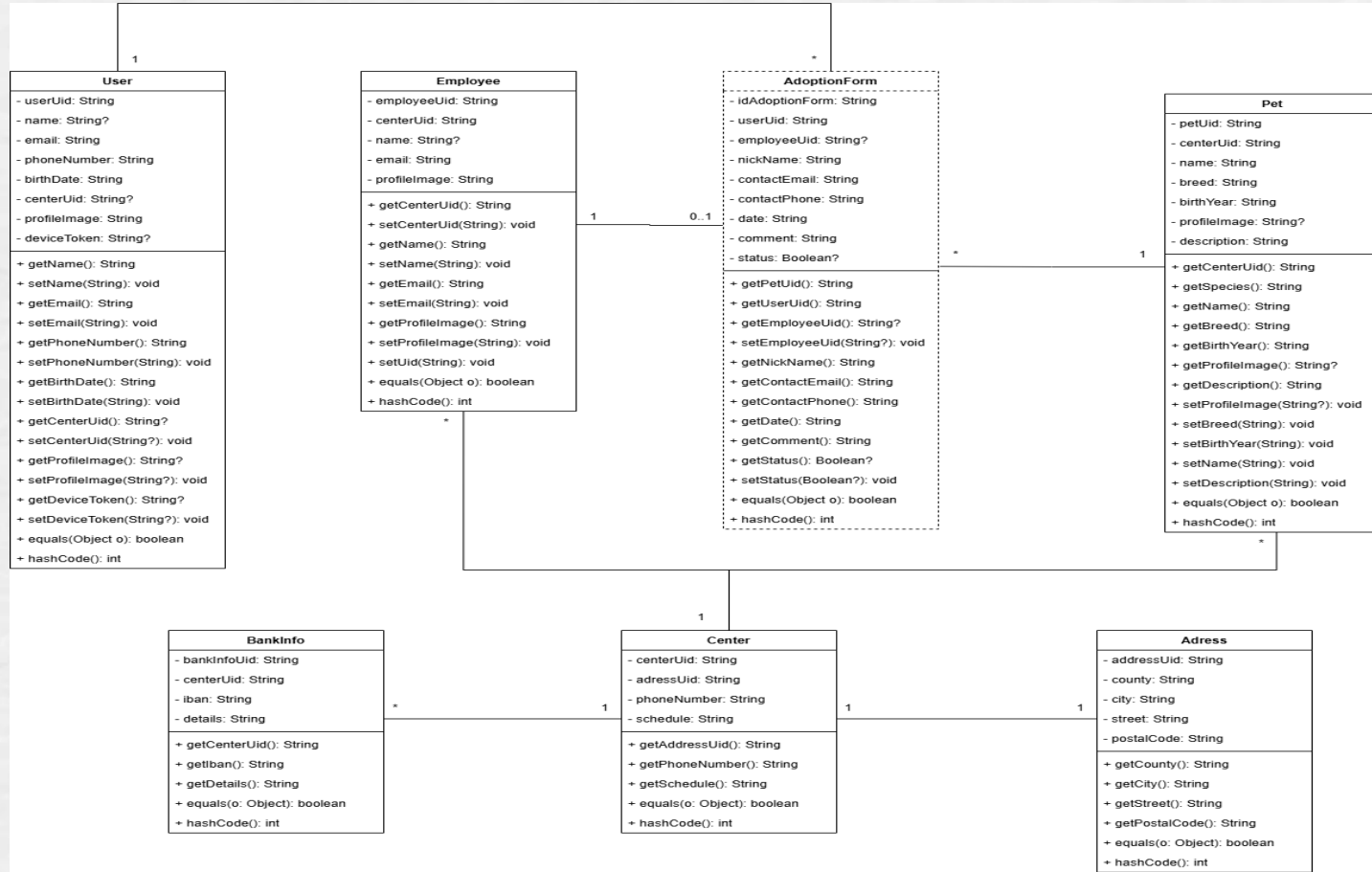




Diagrama de clase



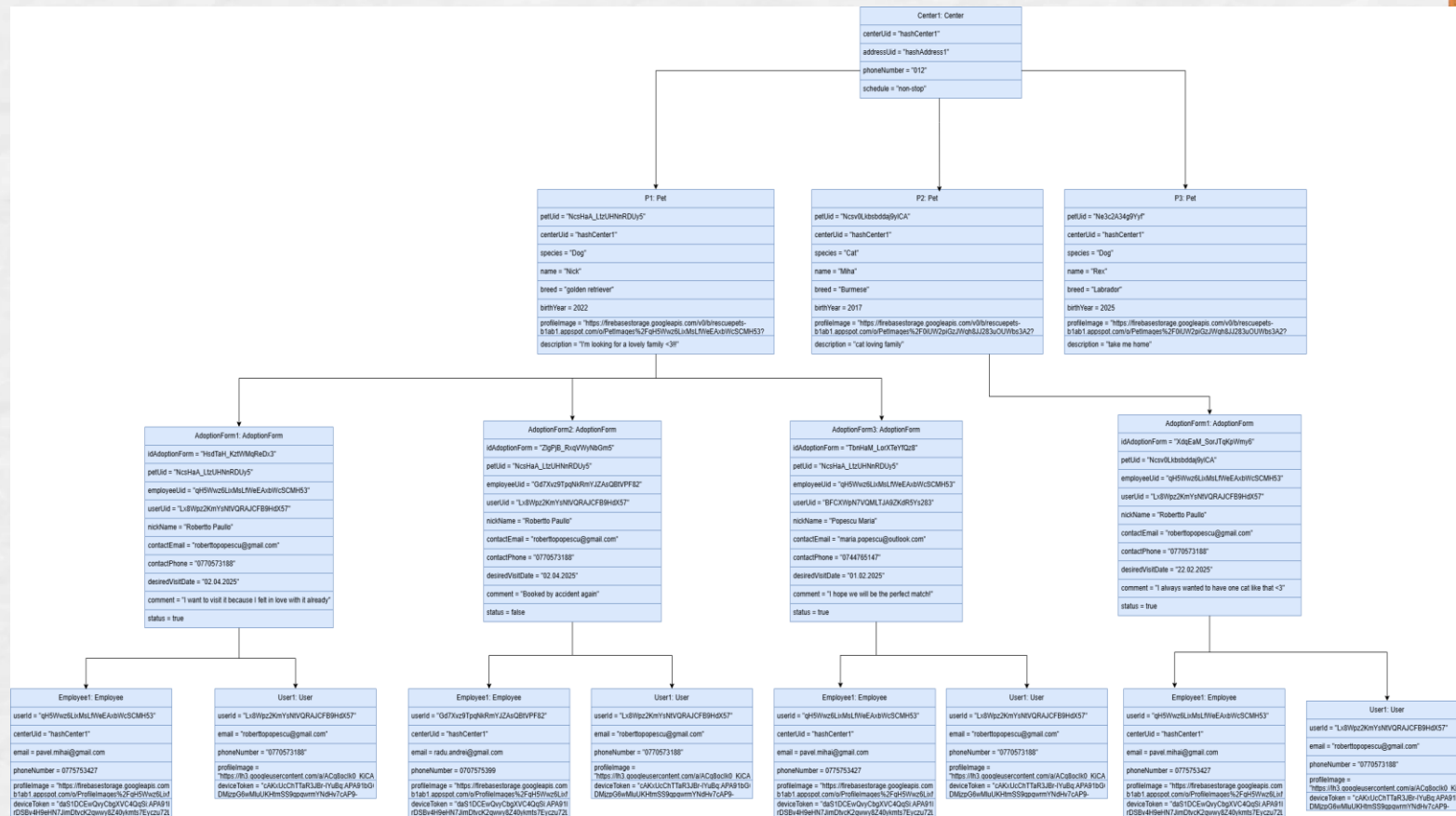
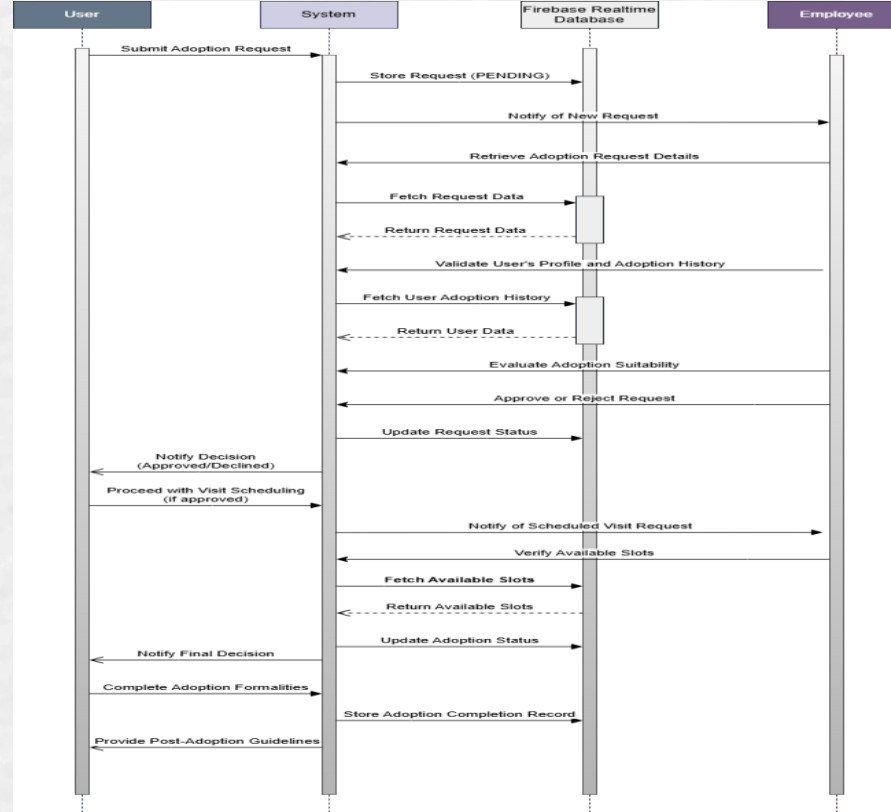




Diagrama de secventa





05 Design patterns



Design patterns



Repository Pattern



Singleton Pattern



Dependency Injection - Dagger



Adapter Pattern



Demo-ul aplicatiei



Multumesc!



Intrebari?
Popescu Paullo-Robertto Karloss

