

# DOCUMENTAȚIA PROIECTULUI

## Coffee Picker

STUDENȚI  
Țițescu Andrei  
Țițescu Gabriel

## Prezentarea proiectului

Acest proiect este o aplicaţie dedicată iubitorilor de cafea. Aplicaţia permite utilizatorilor să descopere diferite tipuri de cafea, să îşi creeze conturi şi să se autentifice pentru a avea acces la funcţionalităţile aplicaţiei.

### Scopul aplicaţiei

Scopul principal al aplicaţiei este de a oferi utilizatorilor o platformă centralizată unde pot explora diferite tipuri de cafea. De asemenea, aplicaţia vizează să ofere informaţii detaliate despre diversele cafele disponibile, inclusiv originea, notele de degustare şi alte detalii relevante. Într-o viitoare dezvoltare, aplicaţia ar permite utilizatorului conectarea wireless la un aparat de cafea compatibil pentru prepararea cafelei dorite.

### Probleme rezolvate

- **Lipsa unei platforme centralizate pentru iubitorii de cafea:** Utilizatorii nu au un loc unic unde să găsească informaţii despre diverse tipuri de cafea.
- **Acces limitat la informaţii detaliate despre cafea:** Informaţiile despre originea şi caracteristicile diferitelor tipuri de cafea sunt adesea fragmentate şi greu de găsit într-un singur loc.

## Tehnologii folosite

### Android Jetpack

Este un set de componente Android dezvoltate pentru a ajuta la construirea aplicaţiilor robuste şi gestionabile. În acest proiect, am folosit următoarele componente Jetpack:

- **Navigation Component:** Utilizat pentru a gestiona navigarea între fragmente şi activităţi.
- **Lifecycle:** Permite gestionarea ciclului de viaţă al componentelor Android, ajutând la prevenirea scurgerilor de memorie.

### Firebase

**Firebase Authentication:** Utilizat pentru autentificarea utilizatorilor. 'FirebaseAuth' oferă autentificare uşoară prin email şi parolă. **FirebaseApp:** Este utilizat pentru iniţializarea Firebase în aplicaţie.

### Room Database

Face parte din Android Jetpack, oferind o interfaţă mai robustă pentru gestionarea bazelor de date locale. În acest proiect, 'AppDatabase' şi 'CoffeeDao' sunt folosite pentru a defini baza de date şi operaţiunile de acces la date. 'DatabaseSingleton' foloseşte acest pattern pentru a asigura existenţa unei singure instanţe a bazei de date în întreaga aplicaţie.

### Shared Preferences

Aceasta tehnologie e utilizata pentru stocarea setărilor aplicaţiei şi a stării aplicaţiei într-o manieră persistentă. În acest proiect, este folosit pentru a verifica dacă datele iniţiale au fost inserate (isDataInserted) şi pentru a gestiona starea de autentificare (isLoggedIn).

## Backend-ul

Partea de backend a aplicaţiei se concentrează pe gestionarea datelor şi autentificarea utilizatorilor, folosind două tehnologii principale: Firebase pentru autentificare şi Room pentru gestionarea bazei de date locale.

### Firebase Authentication

Firebase Authentication este utilizat pentru autentificarea utilizatorilor.

```
if (FirebaseApp.getApps( context: this).isEmpty()) {  
    FirebaseApp.initializeApp( context: this)  
}  
  
auth = FirebaseAuth.getInstance()
```

Această configurare permite autentificarea utilizatorilor prin email şi parolă. De exemplu, la login sau înregistrare, utilizatorii sunt verificaţi şi autentificaţi prin Firebase.

### Room Database

Room Database este utilizată pentru gestionarea datelor locale. Este o componentă a Android Jetpack şi oferă o interfaţă robustă pentru manipularea bazelor de date SQLite.

Este utilizat un singleton pentru a asigura existenţa unei singure instanţe a bazei de date în întreaga aplicaţie.

DAO-ul 'CoffeeDao' defineşte metodele de acces la date:

- 'getCoffeesByType': Returnează o listă de obiecte 'Coffee' de un anumit tip din baza de date.
- 'insertAllCoffees': Inserează o listă de obiecte 'Coffee' în baza de date, înlocuind intrările existente în caz de conflict.

Datele iniţiale sunt inserate în baza de date doar dacă acestea nu au fost deja introduse. Aceasta metodă foloseşte corutine pentru a insera datele în mod asincron.

## Arhitectura

Aplicaţia se bazează pe o arhitectură modulară şi componentizată, utilizând diverse tehnologii şi concepte moderne pentru a asigura o structură robustă şi scalabilă.

### Componente principale

- Ca activităţi avem ‘MainActivity’ care reprezintă punctul de intrare al aplicaţiei, gestionând autentificarea şi navigarea iniţială şi ‘SecondActivity’ care oferă interfaţa principală a aplicaţiei după autentificare.
- Fragmentele aplicaţiei sunt: ‘LoginFragment’ pentru autentificare, ‘RegisterFragment’, unde se face înregistrarea de noi utilizatori, ‘CoffeesFragment’ care face afişarea cafelelor în funcţie de tip, utilizând un RecyclerView şi ‘CoffeeDetailsFragment’ unde se afişează detaliile unei cafele.
- ‘CoffeesAdapter’ este un adaptor pentru RecyclerView şi este responsabil pentru afişarea elementelor din lista de tipuri de cafea.

### Navigare

Navigarea în aplicaţie este gestionată de componentele Navigation Architecture Component ale Android. Fişierul nav\_graph.xml defineşte fluxul de navigare între fragmente şi activităţi:

- LoginFragment este destinaţia iniţială a aplicaţiei. De aici se face navigarea către RegisterFragment sau către SecondActivity dacă autentificarea are succes.
- RegisterFragment oferă o acţiune pentru a reveni la LoginFragment după înregistrare.

### Baza de date

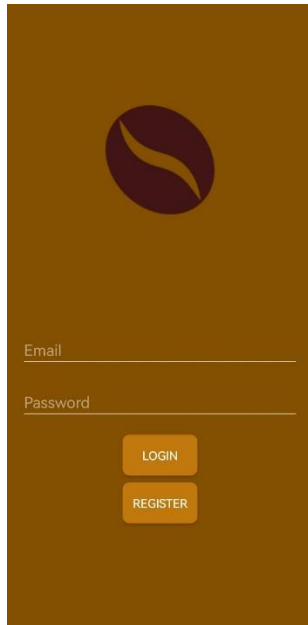
Baza de date utilizată este Room, o bibliotecă de persistenţă oferită de Android, care oferă o abstracţie peste SQLite.

- DatabaseSingleton este o clasă care asigură că există o singură instanţă a bazei de date AppDatabase, utilizând un model de singleton thread-safe.

- AppDatabase: Defineşte baza de date Room, indicând entităţile (în acest caz, Coffee) şi DAO-urile asociate.
- CoffeeDao: Definiţiile DAO includ metode pentru inserarea şi interogarea datelor din tabela ‘coffees’.

## Interfaţă

Interfaţa este construită folosind ConstraintLayout pentru a asigura un layout flexibil şi adaptabil.



MainActivity conţine:

- Conţine un ‘FragmentManager’ care găzduieşte ‘NavHostFragment’ pentru gestionarea navigării.
- Un ImageView pentru afişarea unei imagini reprezentative (un bob de cafea).

În SecondActivity se regăsesc:

- Un ‘FragmentManager’ pentru găzduirea fragmentelor.
- Un ImageButton ce foloseşte drawable selector pentru logout şi un TextView pentru titlu.

CoffeesFragment foloseşte:

- Un RecyclerView pentru afişarea listei de cafele.
- Un LinearLayout cu trei butoane pentru filtrarea cafelelor (hot, cold, to-go).



## Utilizarea aplicaţiei

Aplicaţia de management al cafelelor este concepută pentru a oferi utilizatorilor o experienţă uşor de utilizat, permiţându-le să vizualizeze şi să gestioneze diferite tipuri de cafea. Aplicaţia include funcţionalităţi de autentificare şi înregistrare, asigurând accesul securizat la funcţionalităţile interne.

Un utilizator are de ales fie să se conecteze la un cont existent, fie să creeze un nou cont. Noul utilizator:

- Aplicaţia porneşte la ecranul de autentificare.
- Utilizatorii noi care nu au un cont pot selecta opţiunea de înregistrare, care îi va redirecţiona către ecranul de înregistrare.
- Aici, utilizatorii vor completa detaliile necesare (e.g., email, parolă) pentru a crea un cont nou.
- După înregistrare cu succes, utilizatorii vor fi redirecţionaţi automat la ecranul de autentificare pentru a se autentifica cu noile credenţiale.

Utilizatori existenţi:

- Utilizatorii existenţi îşi vor introduce email-ul şi parola în ‘LoginFragment’ pentru a se autentifica.
- După autentificare cu succes, utilizatorii vor fi redirecţionaţi către ‘SecondActivity’, unde vor avea acces la funcţionalităţile aplicaţiei.

Ecranul principal:

- În partea de sus a ecranului, utilizatorii vor vedea un titlu ("Coffee Menu") şi un buton de logout.
- Butonul de logout permite utilizatorilor să se delogheze şi să revină la ecranul de autentificare.
- ‘CoffeesFragment’ conţine un RecyclerView care afişează tipurile de cafea.
- Fiecare element din listă reprezintă un tip de cafea, cu o imagine reprezentativă şi detalii relevante.
- Sub RecyclerView, există trei butoane de filtrare (Hot Coffees, Cold Coffees, To-Go Coffees) care permit utilizatorilor să filtreze tipurile de cafea afişate în listă.

RecyclerView şi adapteri:

- RecyclerView din CoffeesFragment utilizează un adaptor pentru a lega datele la elementele UI.
- Fiecare element din RecyclerView trimite către o pagină cu informaţii şi detalii relevante despre cafeaua selectată.

## Concluzii

Această aplicaţie de management al cafelelor reuşeşte să ofere o soluţie intuitivă şi eficientă pentru utilizatorii care doresc să exploreze şi să gestioneze diverse tipuri de cafea. Printr-o combinaţie de navigare bine structurată, interfaţă prietenoasă şi funcţionalităţi robuste, aplicaţia asigură o experienţă plăcută şi fără întreruperi pentru utilizatori.

## Puncte forte

- Procesul de autentificare şi înregistrare este simplu şi sigur, protejând datele utilizatorilor şi asigurând accesul controlat la funcţionalităţile aplicaţiei.
- Structura de navigare utilizată permite o trecere uşoară între diferitele secţiuni ale aplicaţiei, făcând experienţa utilizatorului mai fluidă.
- Design-ul aplicaţiei, incluzând elemente vizuale atractive şi layout-uri bine gândite, contribuie la o interacţiune uşoară şi plăcută pentru utilizatori.
- Utilizarea RecyclerView pentru afişarea listelor de cafea permite o vizualizare eficientă şi organizată a informaţiilor.

## Aspecte de îmbunătăţit

- Extinderea funcţionalităţilor: Adăugarea de noi funcţionalităţi, cum ar fi posibilitatea de a adăuga recenzii sau de a marca cafelele preferate, ar putea creşte atractivitatea şi utilitatea aplicaţiei.
- Personalizarea experienţei utilizatorului: Integrarea unor opţiuni de personalizare, cum ar fi preferinţe de afişare, ar putea oferi utilizatorilor o experienţă mai personalizată şi adaptată nevoilor lor individuale.