**Odmevko**

Nina Plevnik, 63210251

Tilen Smrdel, 63210304

# Opis projekta

Odmevko je mobilna aplikacija za predvajanje glasbe, namenjena preprostemu in dostopnemu pretakanju glasbe. Aplikacija omogoča uporabnikom iskanje skladb, upravljanje predvajalnih seznamov ter predvajanje glasbe neposredno iz strežnika. Cilj je ustvariti intuitiven uporabniški vmesnik, ki omogoča hitro in udobno glasbeno izkušnjo.

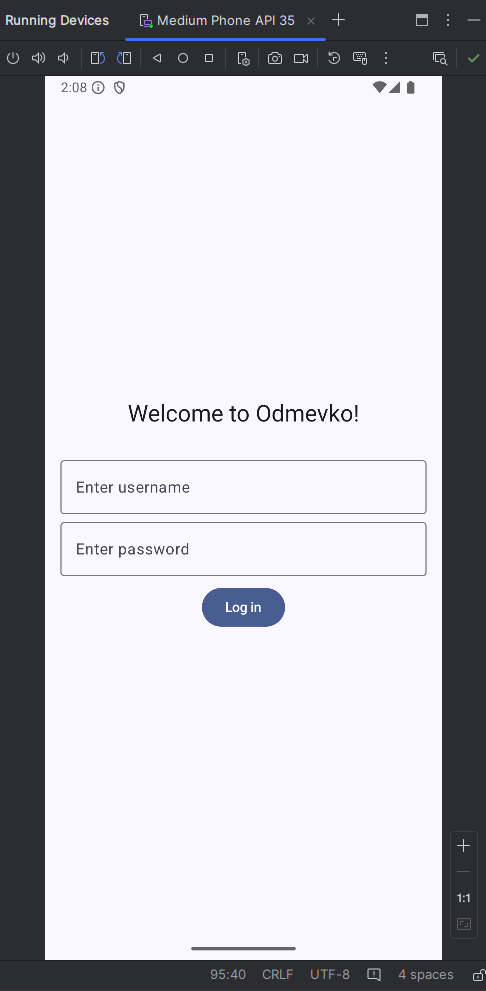
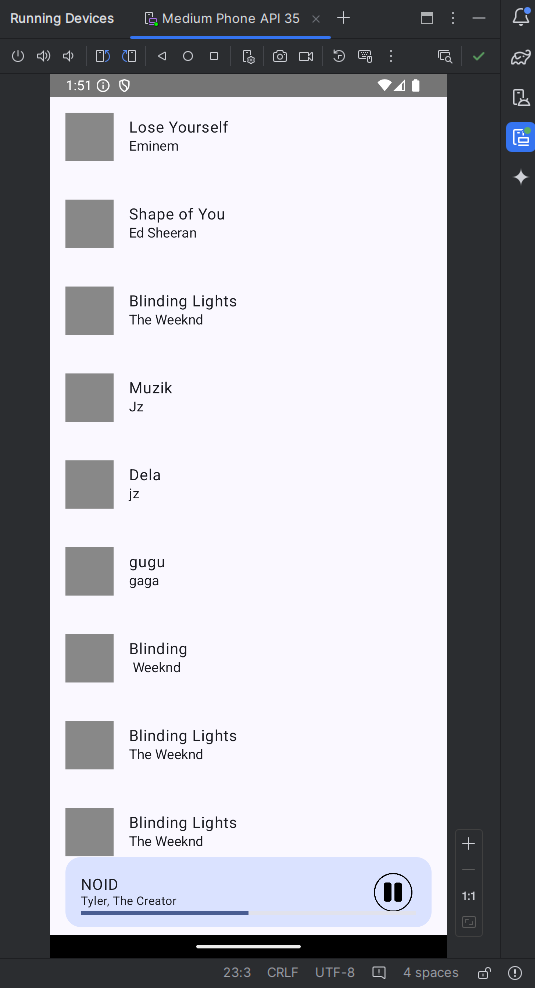
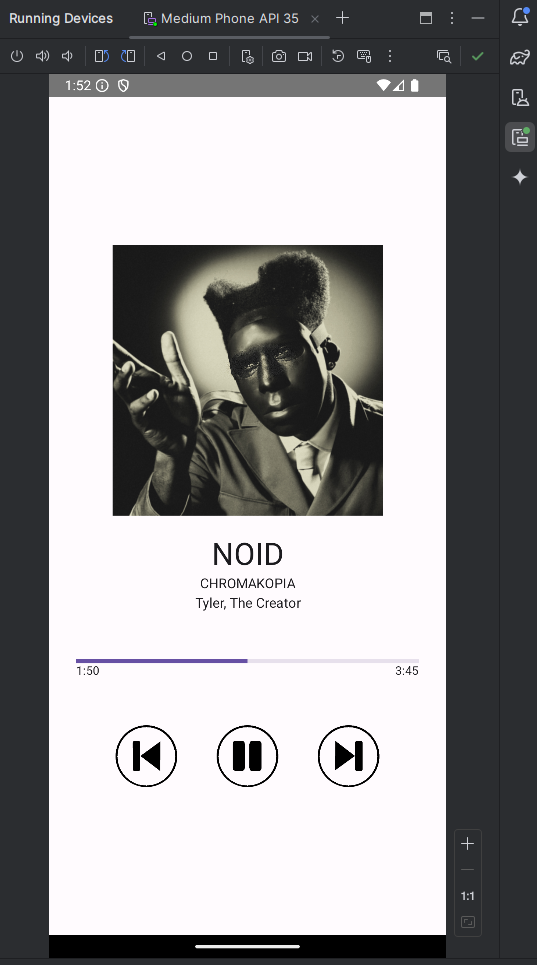
# Tehnologije

* Jezik: Kotlin
* Okvir: Android SDK in Jetpack Compose
* API: Deezer API (prek platforme RapidAPI)
* Knjižnice: Retrofit, Gson, Coil (za nalaganje slik)
* Razvojno okolje: Android Studio

# Povezava do repozitorija na GitHubu

https://github.com/niny8888/EMP

# Trenutno implementirane funkcionalnosti

* Začetni zaslon(Nina Plevnik)
  + To je prvi zaslon s katerim se spozna uporabnik pred uporabo najine aplikacije Odmevko. Noter vpiše svoje uporabniško ime in geslo ter se prijavi v aplikacijo. Namen tega zaslona je avtentikacija uporabnika za dostop do njegove zbirke glasbe
  + 
* Zaslon za zbiranje glasbe (Nina Plevnik)
  + Playlist je seznam vse uporabnikove glasbe, med katero lahko izbira, ob kliku na izbran pesem se v spodnjem delu ekrana prikaže polje, ki prekrije del playlista in prikazuje trenutno pesem in njen status igranja.
  + 
* Zaslon za predvajanje glasbe (Tilen Smrdel)
  + To je glavni zaslon preko katerega uporabnik nadzira predvajanje glasbe iz dosegljive zbirke. Tu vidi tudi album art, čas trajajna komada ki se trenutno predvaja, preskok na naslenjega, preskok na prejšnjega
  + 

# Implementirane funkcionalnosti in primeri uporabe

* **Iskanje skladb:** Uporabnik vnese ime izvajalca ali skladbe in aplikacija prikaže rezultate.
* **Predvajanje glasbe:** Možnost upravljanja predvajanja (predvajaj/premor, naslednja/prejšnja skladba).
* **Seznam predvajanja:** Prikaz skladb z možnostjo izbire za predvajanje.
* **Integracija API-ja:** Pridobivanje podatkov o skladbah, izvajalcih in albumih prek Deezer AP

# Arhitekturna zasnova aplikacije

Aplikacija temelji na **MVVM** arhitekturi:

* **Model:** Razredi za podatkovne modele (npr. Song, Artist, Album, Data).
* **ViewModel:** Upravljanje stanja z uporabo StateFlow in asinhronih klicev API-ja.
* **View:** Zasnova uporabniškega vmesnika z Jetpack Compose in prikaz podatkov iz ViewModel.

# Demonstracija primera izvajanja operacij ob različnih časih v življenjskem ciklu aktivnosti

* **PlaylistActivity.kt:** Aktivnost prikaže seznam skladb. Ob vstopu v aktivnost (onCreate) se inicializira vsebina in vmesnik.
* **SongActivity.kt:** Aktivnost za predvajanje skladb. Prikazuje napredek in informacije o trenutni skladbi.

# Uporaba ViewModel, UI State in StateFlow

* **ViewModel:** Upravljanje poslovne logike in asinhronih API klicev (npr. MusicViewModel).
* **StateFlow:** Pretok stanj, ki zagotavlja reaktivne posodobitve UI ob spremembah podatkov (npr. \_musicData, \_error).
* **UI State:** Komponente (Compose) se posodabljajo na podlagi zbranih stanj (collectAsState).

# Uporaba zunanjih API-jev v aplikaciji

Aplikacija uporablja Deezer API za iskanje glasbe. Zahteve so izvedene prek knjižnice Retrofit, kar omogoča enostavno integracijo in obdelavo podatkov JSON.

# Podatkovni model (shema) za aplikacijo in opis podatkovnega modela

Podatkovni model vključuje naslednje entitete:

* **Artist (Izvajalec):** Informacije o izvajalcu, kot so ime, slike, povezava do skladb (npr. Artist.kt)​Artist.
* **Album:** Podatki o albumu, kot so naslov, naslovnica, povezava do skladb (npr. Album.kt)​.
* **Data:** Vsebuje podatke o posamezni skladbi, povezavi do predogleda in izvajalcu (npr. Data.kt)​.
* **MyData:** Glavni model za odgovore iz API-ja, ki vključuje seznam skladb, število rezultatov in povezavo za nadaljevanje (npr. MyData.kt)​.