BME AUT Automatizálási és Alkalmazott Informatikai Tanszék

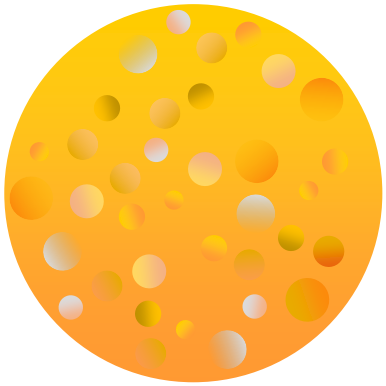
Házi feladat dokumentáció

Android alapú szoftverfejlesztés 2017 tavasz

Antares

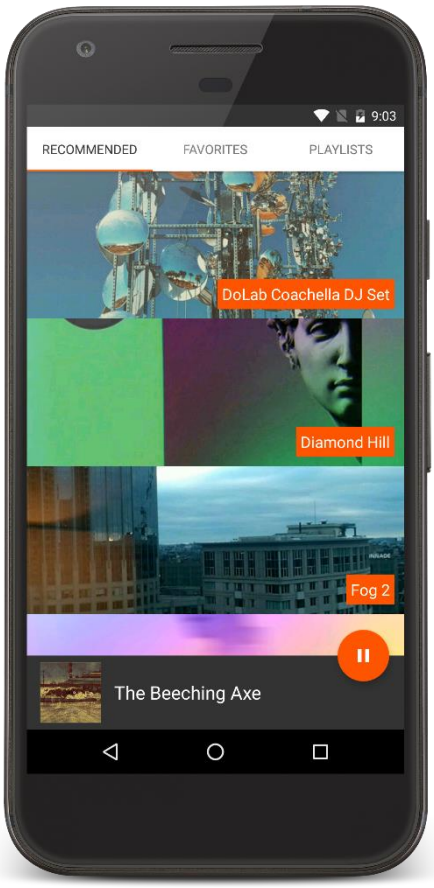
Készítette: Pintér Benedek

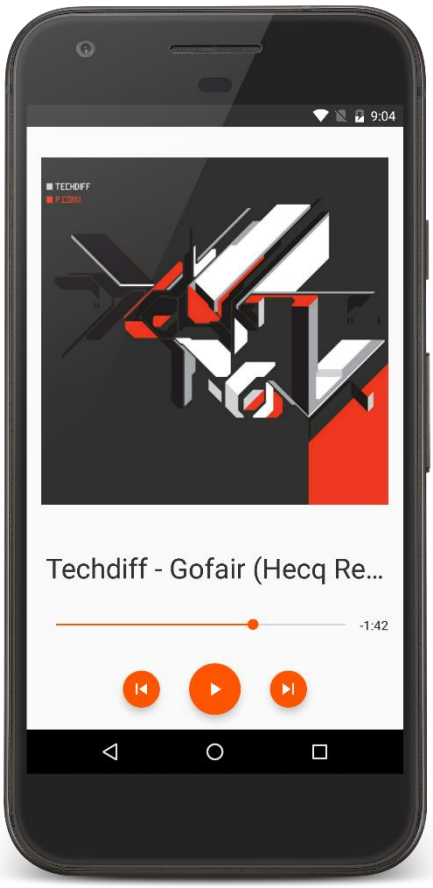
Laborvezető: Horváth János



Bemutatás

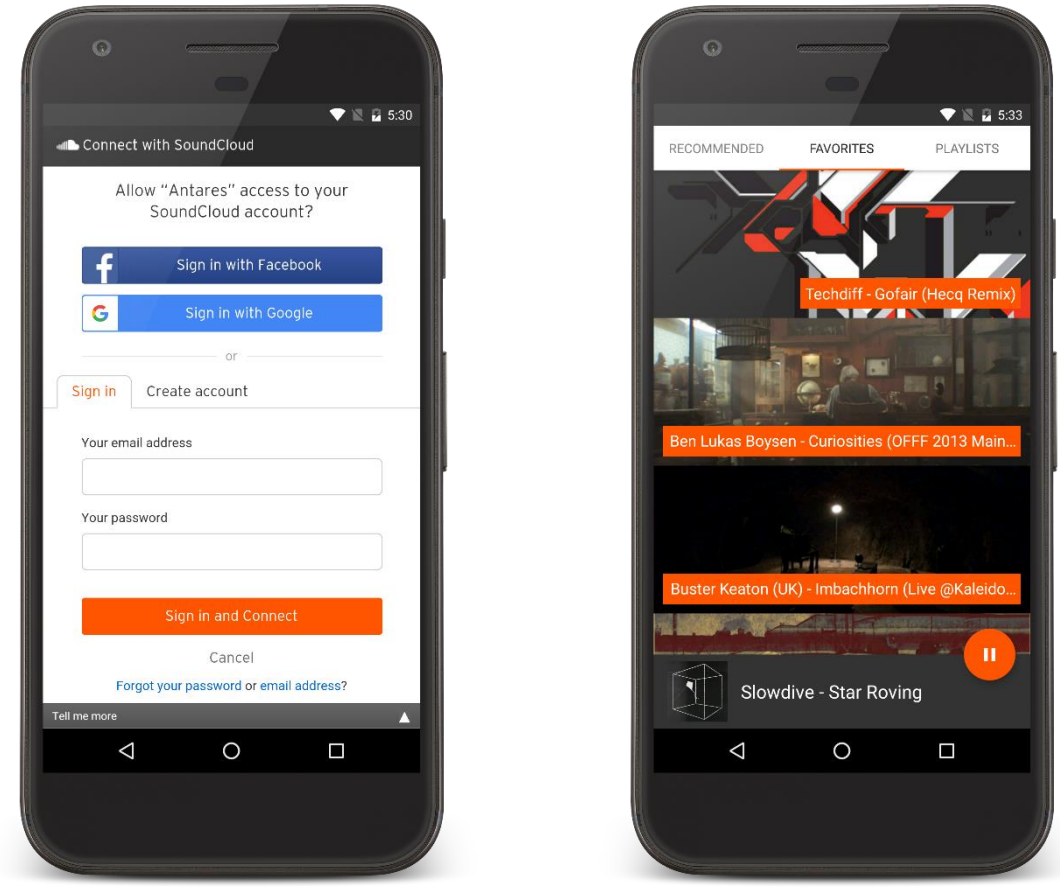
Az Antares egy SoundCloud kliens, amivel a saját kedveléseinket, lejátszási listáinkat, illetve a nekünk ajánlott számokat játszhatjuk le, akár a háttérben futtatva.



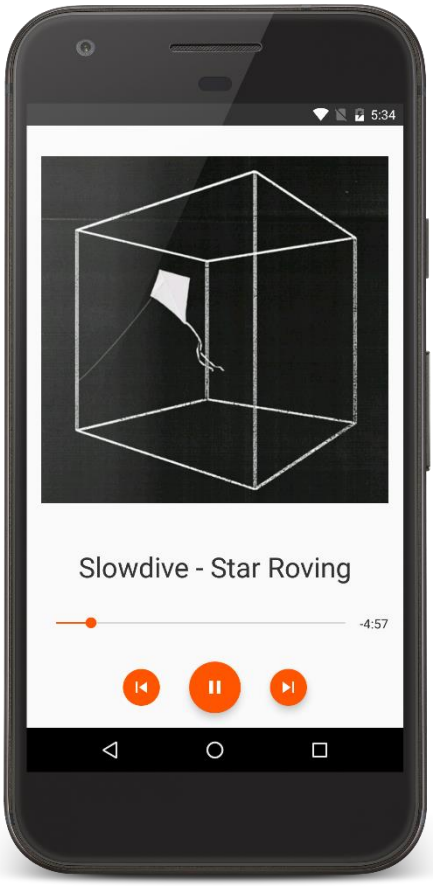


Felhasználói kézikönyv

Első indításkor az alkalmazáson belül be kell jelentkeznünk a SoundCloud fiókunkba, hogy jogosultságot adjunk a saját tartalmunk elérésére. Ezek után a fő képernyő fogad, ahol tabos elrendezésben, három listában láthatjuk a nekünk ajánlott számokat, a saját kedveléseinket, illetve a saját lejátszási listáinkat. Mindegyik lista automatikusan tölti be a további tartalmat interneten keresztül, ahogy közeledünk a lista végéhez, illetve támogatja a „swipe to refresh" funkciót. Ha kiválasztunk egy elemet, akkor azt elkezdi lejátszani a telefon, és alulról beúszik a mini vezérlő, amin keresztül szüneteltethetjük/folytathatjuk a lejátszást, értesülhetünk a bufferelésről, illetve láthatjuk a lejátszott szám címét és képét. A címet vagy a képet megérintve elnavigálhatunk a teljes képernyős lejátszóhoz, ahol szintén megjelenik a kép és cím, továbbá egy csúszkán tekerhetünk a számon belül, láthatjuk a fennmaradó időt, illetve előre/hátra léphetünk a számok között és szüneteltethetjük/folytathatjuk a lejátszást. A zene akkor sem szünetel, ha az alkalmazás a háttérbe kerül. Az ajánlott számok és a kedveléseink esetében a listaelemekből készül playlist, míg a saját lejátszási listáink esetén a kiválasztott lista elemei fognak lejátszódni.



1. ábra: A bejelentkező képernyő és a főmenü, ahol a különböző kategóriákba rendezett számok között válogathatunk. Az alsó sávban látszik a mini lejátszó vezérlő.



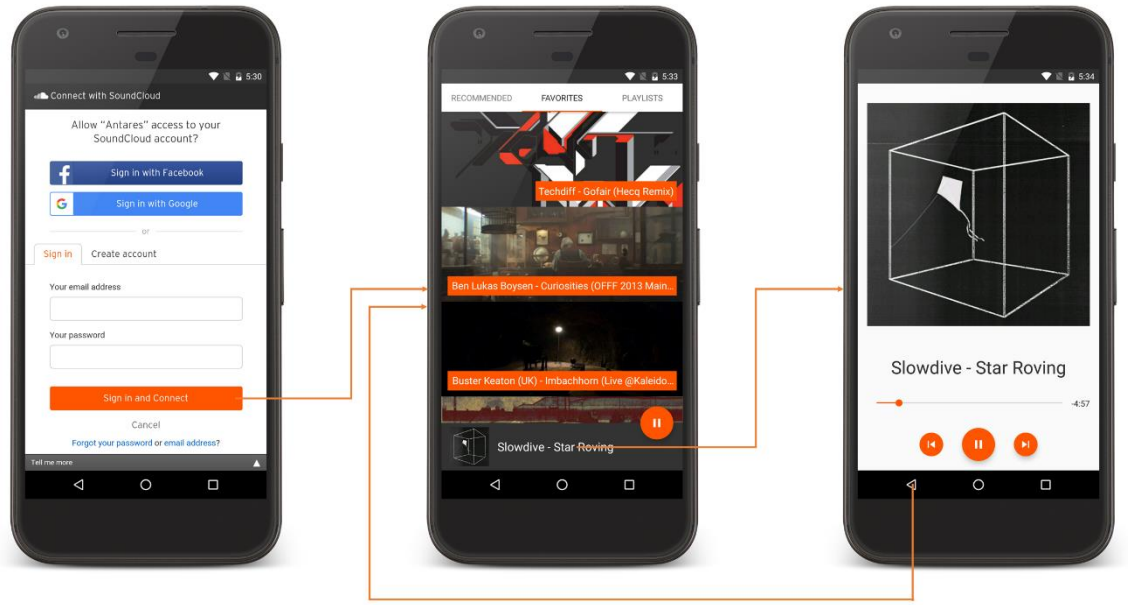
2. ábra: A lejátszó képernyő, ahol tekerhetünk a számon belül és a számok között, illetve leállíthatjuk/folytathatjuk a lejátszást.

Felhasznált technológiák

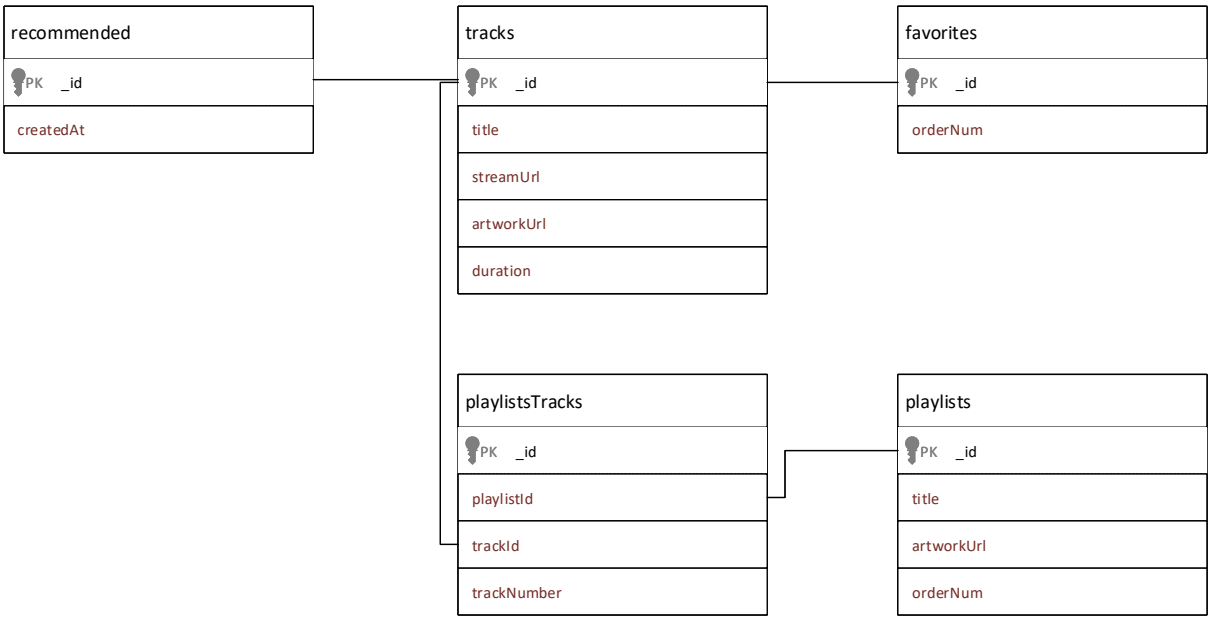
- **WebView** használata a bejelentkezéshez (csak így lehetséges user code-ot kapni)
- **OAuth token automatikus frissítés** minden hálózati kommunikáció előtt, ha az lejárt
- **TabLayout Fragmentekkel**
- A mini playerBar egy **CustomView, egyedi animációk**kal
- **Végtelenített RecyclerView**-k, Swipe to refresh
- Retrofit használata a SoundCloud API használatához
- EventBus-on keresztüli kommunikáció a hálózati modul és az Activity-k, Fragmentek között
- **SQLite** adatbázisba cachelés és **ContentProvider**rel absztrahált hozzáférés, **CursorLoader** használata
- **MediaPlayer** használata külön **Service**-ben (**Bound/Unbound**), funkciók: playlist, play, pause, previous, next, seek
- ButterKnife, **SharedPreferences, WifiLock** használata
- Képek cachelése Glide-al
- Zene bufferelés jelzése FABProgressCircle osztálykönyvtárral

Fejlesztői dokumentáció

Navigáció



Adatbázis séma



Animációk

Gif az animációról

Az alsó playerBar animációja egyedi: lejátszás indítására alulról beúszik, a FAB rajta pedig egy kis késleltetéssel 0-ról az eredeti méretére skálázódik. A mögötte lévő RecyclerView alját feltolja, hogy ne takarja ki a legutolsó elemet se a playerBar. A FAB körül egy vékony vonal kezd el körbemenni, ha a lejátszó éppen buffereli a zenét, viszont ezt már egy segédkönyvtárral oldottam meg. Az animációk visszafelé, eltünés esetén is működnek.

Érintett osztályok:

- PlayerBarView
- MainActivity
- PlayerActivity

PlayerService

A zene streamelő Service-t igyekeztem minél jobban leválasztani az alkalmazásról, és egy szolgáltatás interfészt kialakítani neki, hogy minél inkább újra felhasználható legyen egy másik alkalmazásban is. A Service egyszerre Bound és Unbound típusú és WifiLockot használ, hogy lezárt képernyő mellett se szakadjon meg a streamelés.

Érintett osztályok:

- PlayerService
- PlayerBinder

Adatelérési réteg

Az adatelérési réteg alapjai le lettek lettek fektetve, innentől kezdve nagyon egyszerű bővíteni az alkalmazást, például egy szám részletes nézetéhez csak a Track modelbe kell felvenni az új változókat és az adatbázis táblához hozzáadni az oszlopokat, semmi más részlettel nem kell foglalkozni, a fő munkát a grafikus felület elkészítése jelentené ebben az esetben. Illetve elég sok kód foglalkozik azzal, hogy a SoundCloud API-tól megkapott adatformátumot átalakítsa az adatbázisban tárolt formátummá, majd ContentProvideren keresztül elérhetővé tegye az alkalmazás felsőbb rétegei számára egy harmadik, kényelmesen kezelhető formátummá.

Érintett osztályok:

- SoundCloudAPI, SoundCloudInteractor, ActivitiesResponse, ActivitiesContent
- TracksTable, RecommendedTable, FavoritesTable, PlaylistsTable, PlaylistsTracksTable, AntaresDb
- DbContentProvider, DbHelper, FavoritesContract, RecommendedContract, PlaylistsContract, PlaylistTracksContract
- Track, Playlist

Osztálydiagram