

Online Bevásárló Lista Alkalmazás

Dokumentáció

Az alkalmazás lényege

Az Online Bevásárló Lista alkalmazás segítségével a felhasználó bármikor, bárhol feljegyezheti a szükséges termékeket az androidos készülékére, így azokat nem kell észben tartania a legközelebbi bevásárlásig. A felhasználó létrehozhatja a saját listáit, és akár feliratkozhat másokéra is, így láthatja a rajta lévő termékeket, és szerkesztheti is azt. Így akár egy közös családi bevásárlási listát is létre lehet hozni, amelyhez mindenki hozzáadhatja a kívánt termékeket, így semmi sem kerül feledésbe. Az alkalmazás online tárolja a listákat, és folyamatosan frissíti őket.

A bevásárlás során egyesével tudjuk kipipálni a listán a termékeket, miközben azokat a kosarunkba rakjuk, majd a végén a kipipált termékeket egyetlen gombnyomással letörölhetjük a listáról, jelezve, hogy azok bevásárlásra kerültek.

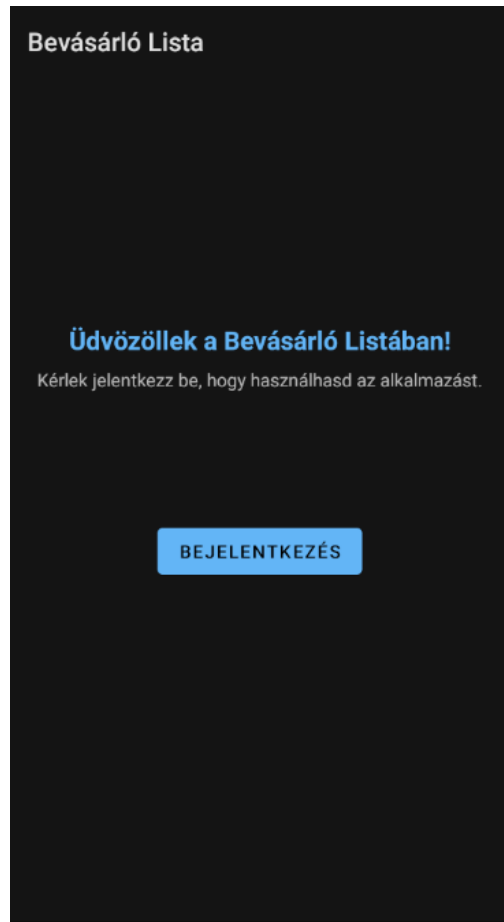
Felhasznált technológiák

Az alkalmazás **Android** platformra, **Kotlin** nyelven készült. Nagybán épít a **Firebase** különböző szolgáltatásaira. A listák és a termékek online tárolását és a készüléken történő offline cachelését a **Firebase Firestore** funkciójával értem el. A felhasználó kezeléshez a **Firebase Authentication** használtam, illetve a bejelentkezési és regisztrációs felületek a **FirebaseUI Authentication** moduljával készültek el. A profilkép online tárolásához a **Firebase Storage**-ot, a megjelenítéséhez a **Glide** külső könyvtárat használtam. A MaterialDialogs nevű népszerű androidos külső könyvtárat használtam a Material megjelenésű, felhasználóbarát dialógusablakok elkészítéséhez.

Az alkalmazás használata

Az alkalmazás használatához a felhasználónak létre kell hoznia egy saját fiókot. Erre azért van szükség, hogy bármilyen készülékről be tudjon jelentkezni, és láthassa a saját listáit, illetve készülékcseré esetén azok ne vesszenek el.

Ha nincs jelenleg bejelentkezett felhasználó, akkor az app indulásakor az üdvözlő képernyő jelenik meg. A regisztrációs és bejelentkeztetési képernyőket a FirebaseUI Authentication moduljával automatikusan hoztam létre. Amennyiben már van bejelentkezett felhasználó, akkor a Firebase Authentication segítségével a felhasználót automatikusan bejelentkezteti az app és egyből a főképernyőre kerül.



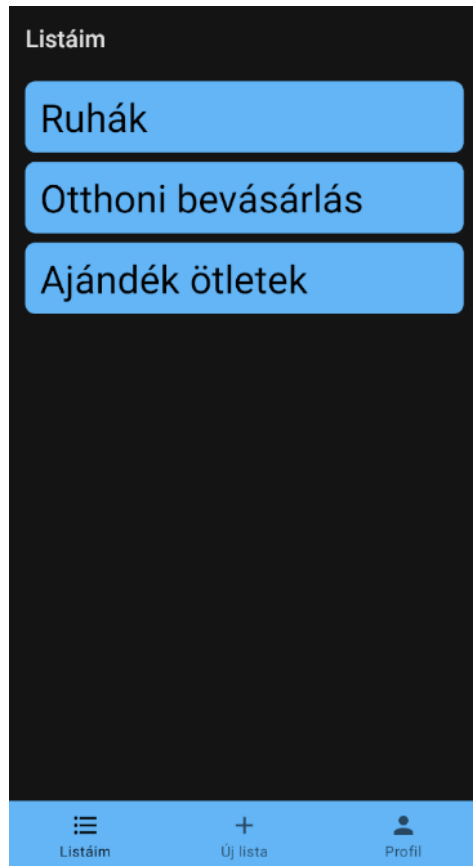
1. ábra - Üdvözlő képernyő

A főképernyőn a felhasználó a képernyő alján található navigációs sáv segítségével választhat a három fő fragmens közül.

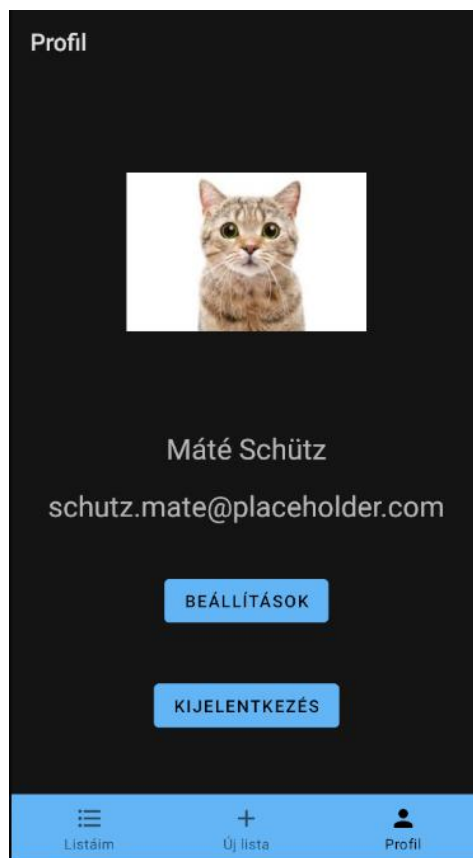
A „Listáim” képernyő azokat a listákat tartalmazza, amelyekre a felhasználó feliratkozott. Egy listára hosszan nyomva leiratkozhatunk a listáról, illetve kimásolhatjuk annak azonosítóját, így azt ismerőseinknek megadva ők is feliratkozhatnak az adott listára.

Az „Új lista” képernyőn új listát hozhatunk létre, amire automatikusan fel is iratkozunk, vagy feliratkozhatunk egy már meglévő listára, annak azonosítójának megadásával.

A „Profil” képernyőn a bejelentkezett felhasználó adatai jelennek meg (email, név, profilkép). Itt a profilképre nyomva beállíthatunk magunknak egy tetszőleget a készülékünkről. A „Beállítások” gombra nyomva megjelenik a „Beállítások” képernyő, ahol az app témáját állíthatjuk be.

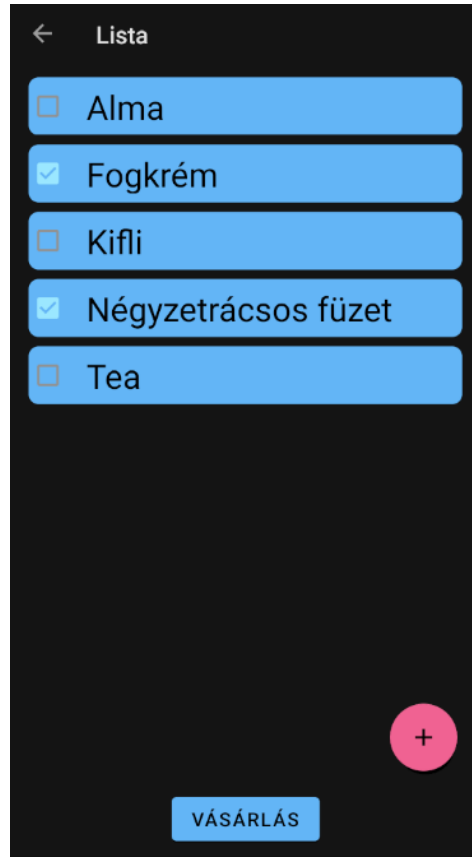


2. ábra - Fő képernyő



3. ábra - Profil képernyő

A „Listáim” képernyőn, egy adott listára nyomva megjelenik a „Lista” képernyő, amely az adott listán lévő termékeket tartalmazza. Egy termékre nyomva, az kipipálódik, és így jelezhetjük, hogy azt vásárlás közben már beraktuk a kosárba. Egy termékre hosszan nyomva átnevezhetjük, vagy akár ki is törölhetjük azt. A jobb alsó sarokban lévő „+” gombra nyomva új terméket adhatunk a listához. A „Vásárlás” gombra nyomva az összes kosárban lévő termék eltűnik a listáról.



4. ábra - Lista képernyő

Architektúra

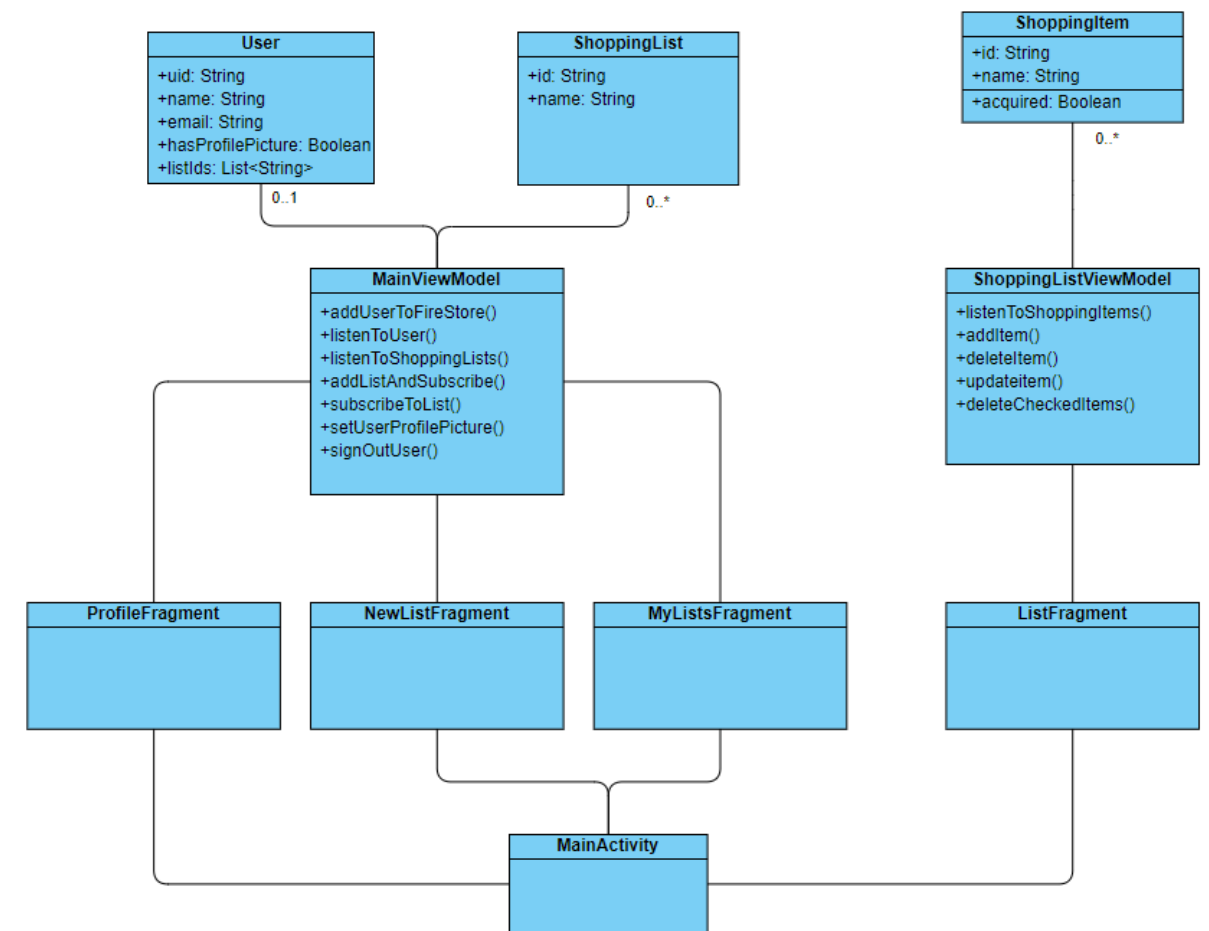
Az alkalmazást a Model-View-ViewModel architektúrát követve építettem fel. Azért döntöttem az MVVM mellett, mivel egyrészt a WPF-es és UWP-s tapasztalataim során ezt az architekturális mintát ismertem meg legjobban, illetve a Google által is ez az ajánlott minta ilyen méretű androidos alkalmazások számára. A következő ábrán látható az alkalmazás osztálydiagramja, amin csak az architektúra szempontjából fontosabb osztályokat tüntettem fel.

Model osztályok: User, ShoppingList, ShoppingItem

ViewModel osztályok: MainViewModel, ShoppingListViewModel

View osztályok: MainActivity, ProfileFragment, NewListFragment, MyListsFragment, ListFragment

Az alkalmazás egy *single-activity Android app*, azaz egyetlen activity-je van (MainActivity), és ezen jelennek meg a különböző fragmensek.



5. ábra - Az alkalmazás architektúrája

Osztályok

Models package:

- **FirestoreShoppingItem**: data class, ami egy Firestore-ban tárolt terméket reprezentál. Azért van szükség rá, mivel a termék azonosítóját a Firestore adja meg automatikusan a terméket reprezentáló dokumentum létrehozásánál, így ebben az osztályban nincs külön azonosító property.
- **FirestoreShoppingList**: data class, ami egy Firestore-ban tárolt bevásárlási listát reprezentál.
- **ShoppingItem**: data class, amely az alkalmazásban tárolt, listában megjelenített terméket reprezentálja. Ebben az osztályban már van azonosító property (id), amely a Firestore-ban tárolt dokumentum egyedi azonosítójának felel meg.
- **ShoppingList**: data class, amely egy, az alkalmazásban lévő bevásárlási listát reprezentál.
- **User**: a felhasználót reprezentáló data class. Tárolja a felhasználó azonosítóját, amelyet a Firebase Authentication hoz létre, valamint a felhasználó nevét, email címét, azt, hogy van-e neki feltöltött profilképe, illetve azon listák azonosítóját, amelyekre a felhasználó feliratkozott.

ViewModels package:

- **MainViewModel**: az alkalmazás leglényegesebb osztálya, a fő képernyőn megjelenő tartalomhoz tartozó ViewModel osztály. A MyListsFragment, a NewListFragment és a ProfileFragment View osztályok adatainak tárolásáért felelős és a hozzájuk tartozó üzleti logikát megvalósító osztály. Tartalmazza a bejelentkezett felhasználóhoz tartozó User

objektumot és a felhasználó bevásárló listáit. Emellett tartalmazza az üzleti logikát a következő funkciókhoz: kommunikálás a Firestore-ral, és a felhasználó adatainak, illetve a listáinak, és az azon lévő termékeknek a frissítése, amennyiben azok változnak a szerveren, új lista hozzáadása, feliratkozás egy listára, leiratkozás egy listáról, profilkép beállítása és feltöltése a Firebase Storage-be, a felhasználó kijelentkeztetése.

- **ShoppingListViewModel:** a ListFragment által megjelenített tartalomhoz tartozó ViewModel osztály. Kommunikál a Firestore-ral és folyamatosan frissíti az adott bevásárló listához tartozó termékeket. A következő funkciókat valósítja meg: új termék hozzáadása, termék törlése, termék frissítése új adatokkal (név, bent van-e a kosárban), összes kosárban lévő termék törlése.
- **ShoppingListViewModelFactory:** Factory osztály, a segítségével hozhatjuk létre a ShoppingListViewModel objektumot.

Views package:

- **ListFragment:** egy adott bevásárló lista termékeit jeleníti meg. A hozzá tartozó üzleti logikát a ShoppingListViewModel osztály tartalmazza, így referenciája van rá. A fragmens tartalmaz egy FloatingActionButton-t, amelyre nyomva új terméket adhat a listához a felhasználó, illetve egy „Bevásárlás” gombot is, amelyre nyomva az összes kosárba tett termék eltűnik a listáról. Egy termékre hosszan nyomva egy dialógus ablak jelenik meg, amelyen átnevezni, illetve kitörölni lehet a terméket.
- **MainActivity:** az app egyetlen activity-je, ez jeleníti meg a különböző fragmenseket, illetve itt történik a fragmensek közötti navigáció a NavController segítségével, és az alkalmazás témájának beállítása is. Tartalmaz egy BottomNavigationView-t amellyel a három fragmens között navigálhat a felhasználó.
- **MyListsFragment:** a felhasználó listáit megjelenítő fragmens. Egy termékre hosszan nyomva a felhasználó kimásolhatja egy lista azonosítóját, illetve leiratkozhat a listáról.
- **NewListFragment:** itt adhat hozzá új listát a felhasználó, illetve feliratkozhat egy már létezőre, az azonosító megadásával.
- **ProfileFragment:** a bejelentkezett felhasználó adatait (email, név, profilkép) megjelenítő fragmens. A profilképre kattintva a felhasználót átnavigálja a készülék galériájába, innen választhat egy képet a felhasználó, amely ezután feltöltődik a Firebase Storage-ba, és megjelenik a fragmens felületén. Emellett a fragmens tartalmaz egy „Beállítások” gombot, amelyre kattintva a felhasználót átnavigálja a SettingsFragment felületére.
- **SettingsFragment:** az alkalmazás beállításait tartalmazó fragmens. Egyelőre csak az app témáját lehet beállítani (világos, sötét, alapértelmezett).
- **WelcomeFragment:** ha nincs bejelentkezett felhasználó, akkor először ez a fragmens jelenik meg. Itt választhat a felhasználó a bejelentkezés és a regisztráció funkciók között. Amint sikeres a bejelentkezés, megjelenik a főképernyő.