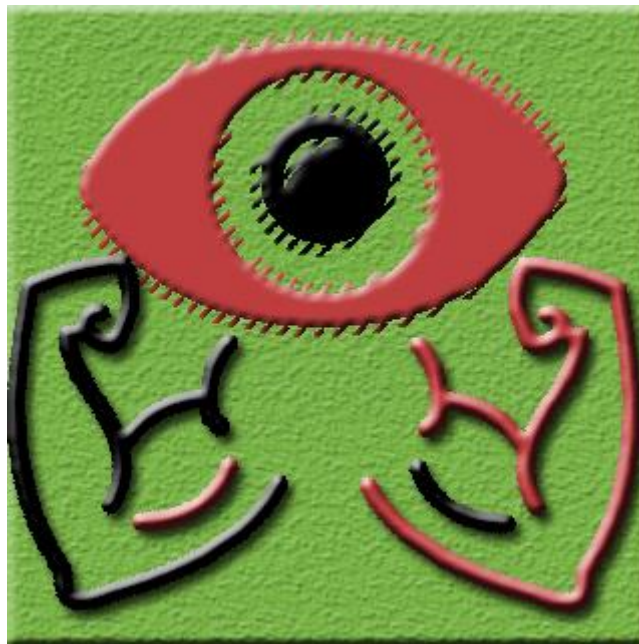


Házi feladat dokumentáció

Android alapú szoftverfejlesztés

[Félév: 2021 őszi]



Tesis Life Buddy

Szabó Gergő - APTEI9

szabo01gergo@gmail.com

Laborvezető: Pásztor Dániel



Bemutató

Az alkalmazással a felhasználó nyomon követheti, hogy a nap folyamán mennyi kalóriát, illetve fehérjét vitt be a szervezetébe. Gyakran fogyasztott ételek (és kalória- fehérjeattribútumaik) elmenthetők az appba, illetve lehetőség van egy internetes adatbázisból és lekérni ételeket. Az ötlet személyes igényből jött, szoktam számolni mind a kalóriát, mind a fehérjét naponta, de könnyebb lenne ezt számoltatni egy dedikált app segítségével. Bár vannak elérhető szoftverek, amelyek ezt a célt szolgálják, a zömük rengeteg plusz funkciót is tartalmaz, amelyek nekem nem szükségesek, és amik átláthatatlanná teszik az egész alkalmazást (plusz az ilyen appok általában a felhasználó arcába nyomják rendszeresen a fizetős verziót). Ezért lett igényem egy letisztult alkalmazásra, amiben a számomra fontos funkciók könnyen elérhetőek.

Főbb funkciók

A felhasználónak lehetősége van számlálókön keresztül nyomon követni, hogy bevitték-e a megfelelő napi mennyiségű kalóriát és fehérjét. Az elvárt napi mennyiségek beállíthatók, illetve a számlálók nullázhatóak.

Kártyaként lementhető ételek egy listába, amely egy elemén látszik az étel neve, kalória- és fehérjeértékei, a mennyiség (ami az adott kalóriát és fehérjét tartalmazza), valamint az adott brand-et (pl. KFC), amihez az étel tartozik (nem minden étel tartozik brandhez). Lehetőség van új kártyát felvenni, törölni, illetve keresni a kártyák között név alapján (ilyenkor a keresett elemek a lista tetejére kerülnek).

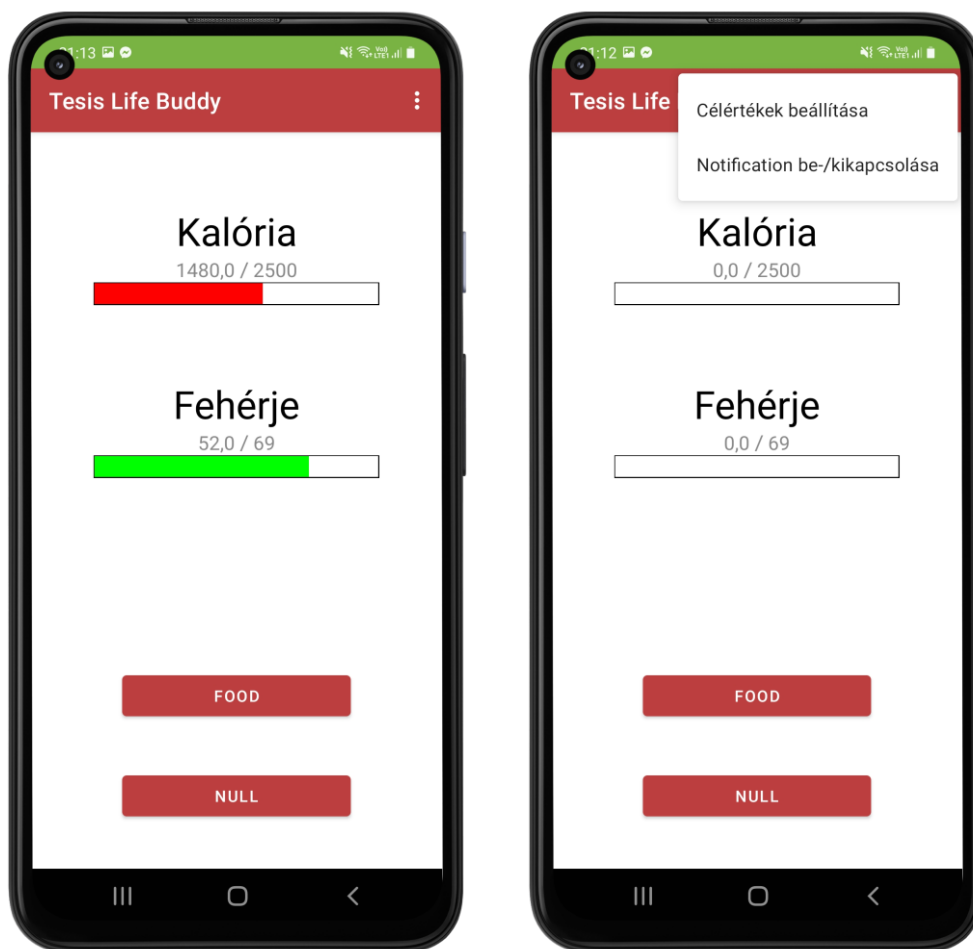
A mentett ételek kalória- és fehérje értékeik hozzáadhatók a számlálókhoz, valamint ezek a számlálók egyénileg (tehát elmentett kártya nélkül) is növelhetőek.

Ezentúl a felhasználó még az internetről is kérhet le ételeket név alapján. Az adatok a [Nutrionix](#) adatbázisából kerülnek letöltésre. A talált ételek kalória- és fehérjeértékei hozzáadhatóak a számlálókhoz, illetve az adott étel kártyaként is lementhető.

Az appban bekapcsolható, hogy megjelenítse-e notificationben a kalória- és fehérjeszámlálók értékeit. Ha ezt bekapcsoljuk, akkor a notification meg lesz jelenítve, egészen addig, amíg ugyanitt, ezt ki nem kapcsoljuk. A notificationre nyomva bejön az alkalmazásunk kezdő képernyője.

Felhasználói kézikönyv

Az alkalmazás elindításakor a felhasználó a kezdőképernyőt láthatja, amely tartalmazza a kalória- és fehérjeszámlálót, két gombot, és egy menüt.

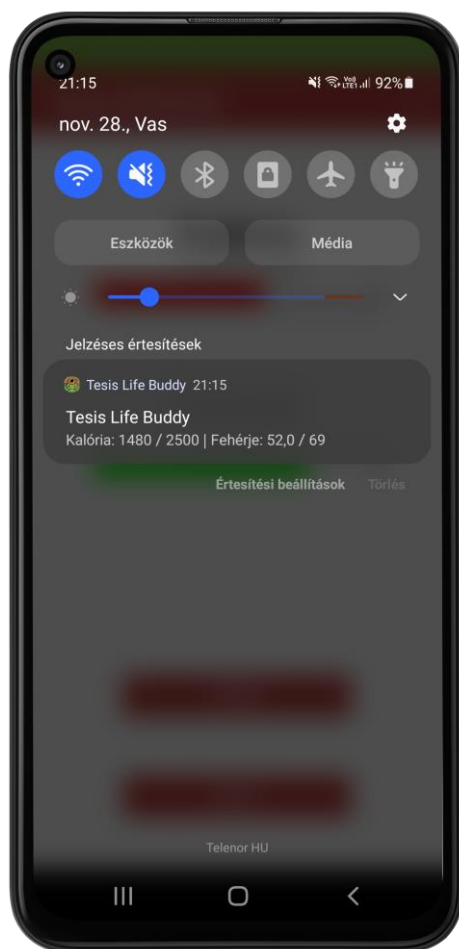


1. Kép: Kezdőképernyő kinézete, menü tartalma

A "Célértékek beállítása" menüpontra kattintva, megjelenik egy dialógusablak, ahol beállíthatók a számlálók célértékei (az 1. képen ez a 2500 és 69-es értékek a célértékek). A dialógusablakban meg kell adni a kért értékeket. Ha úgy nyomunk az "Ok" gombra, hogy valamelyik mezőt üresen hagytuk, akkor az app hibát jelez. Ha úgy nyomunk az "Ok" gombra, hogy ki vannak töltve a mezők, akkor a dialógus bezáródik és a célértékek átállítódnak. Ha a "Cancel" gombra nyomunk, akkor a dialógus bezáródik, más nem történik.

A "Notification be-/kikapcsolása" menüpontra kattintva, ha eddig nem szerepelt notification, akkor megjelenít egyet, ami a számlálók értékeit tartalmazza. Ha újra rányomunk erre a menüpontra, akkor a notification eltűnik.

Megjegyzés: A notification az alkalmazásból való kilépés után is megmarad, akkor tűnik el, ha azt menüből kikapcsoljuk. A notification-re nyomva belépünk az alkalmazás kezdőképernyőjére.



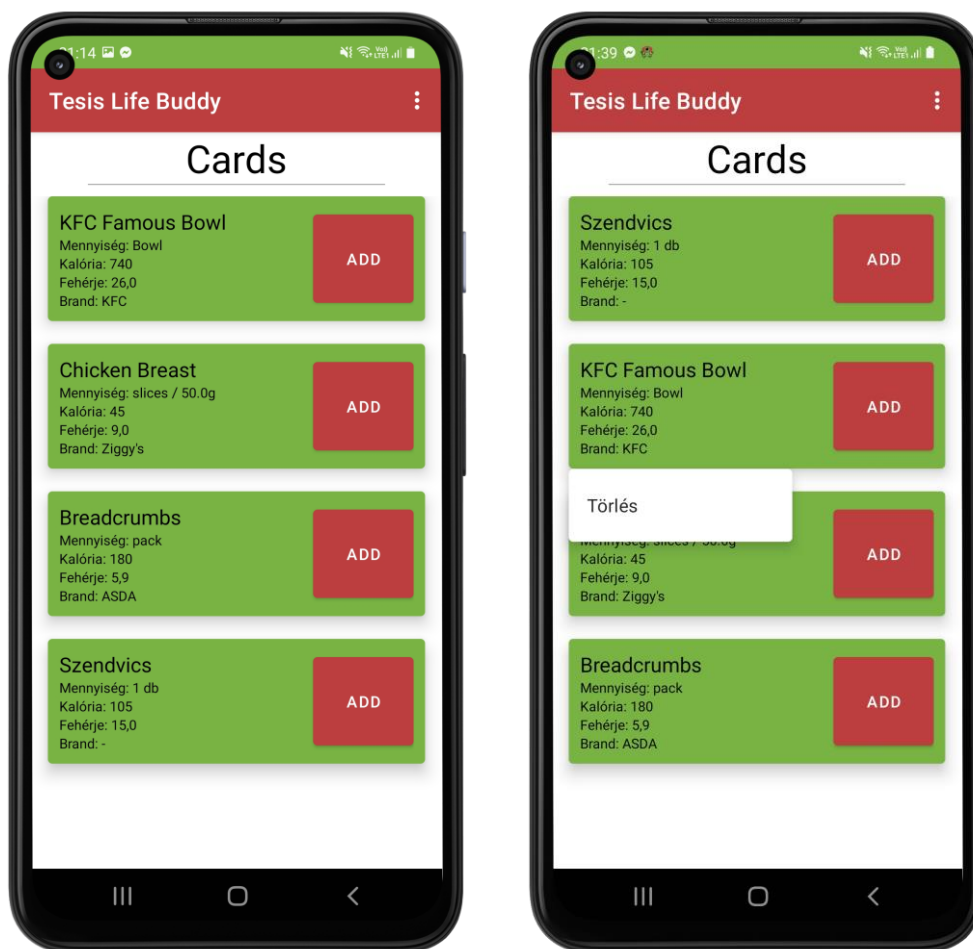


2. Kép: Bal oldalon felül látható a “Célértékek beállítása” menüpont dialógusablaka, bal oldal alul példa arra, hogy milyen amikor egy dialógusablak hiányzó adat miatt hibát jelez. Jobb oldalon pedig a megjelenített notification látható a “Notification be-/kikapcsolása” menüpont eredményeképp

A “Null” gombra kattintva a számlálók értékei (az 1. kép baloldalán az 1480 és az 52 értékek) nullázódnak.

A “Food” gombra kattintva bejön egy új képernyő (továbbiakban: Cards képernyő), ahol a felhasználó az elmentett kártyáit láthatja egy görgethető listában. Minden kártya tartalmazza az étel nevét, kalória- és fehérjeértékeit, a mennyiséget (ami az adott kalóriát és fehérjét tartalmazza), valamint az adott brand-et (pl. KFC), amihez ez étel tartozik, már ha van ilyen (ha nincs brand-je, akkor annak a helyén egy “-” szerepel). Minden listaelemen található továbbá egy “Add” feliratú gomb, aminek a hatására az adott kártya kalória- és fehérjeértékei hozzáadódnak a számlálókhoz.

A kártyák közül elemet úgy törölhetünk, hogy nyomva tartjuk rajta az ujjunkat, ekkor megjelenik egy pop up menü, ahol a “Törlés” menüpontra kattintva az elem törlődik.



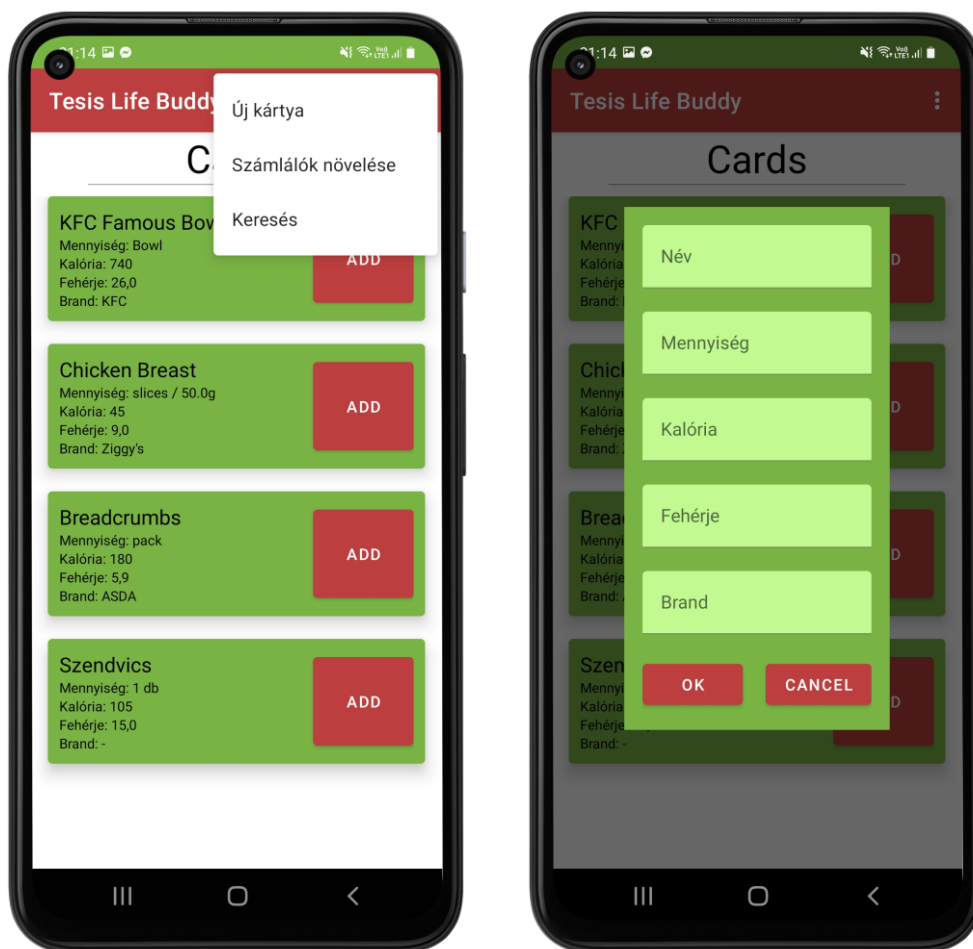
3. Kép: Bal oldalon látható, hogy néz ki a Cards képernyő rajta 4 elemmel, jobb oldalon pedig annak a következménye látható, hogy a második elemen nyomva tartottuk az ujjunkat és megjelent a pop up menü

Ha a felhasználó itt rányom a jobb felső sarokban látható menüre, akkor három menüpont közül választhat:

1) "Új kártya": Erre nyomva megjelenik egy dialógus, ahol lehetőség van új elemet felvenni a kártyák közé. A következő adatokat kell megadni:

- "Név": Az étel neve. Kötelező megadni, maximum 15 karakterből állhat. Szöveg.
- "Mennyiség": A mennyiség, ami majd a megfelelő kalória- és fehérje értékeket tartalmazza. Kötelező megadni, maximum 28 karakterből állhat. Szöveg.
- "Kalória": A tartalmazott kalória értéke (kcal-ban). Kötelező megadni. Egész vagy tört szám.
- "Fehérje": A tartalmazott fehérje értéke (gramban). Kötelező megadni. Egész vagy tört szám.
- "Brand": A brand, amihez az étel tartozik. Nem kötelező megadni, maximum 32 karakterből állhat. Szöveg.

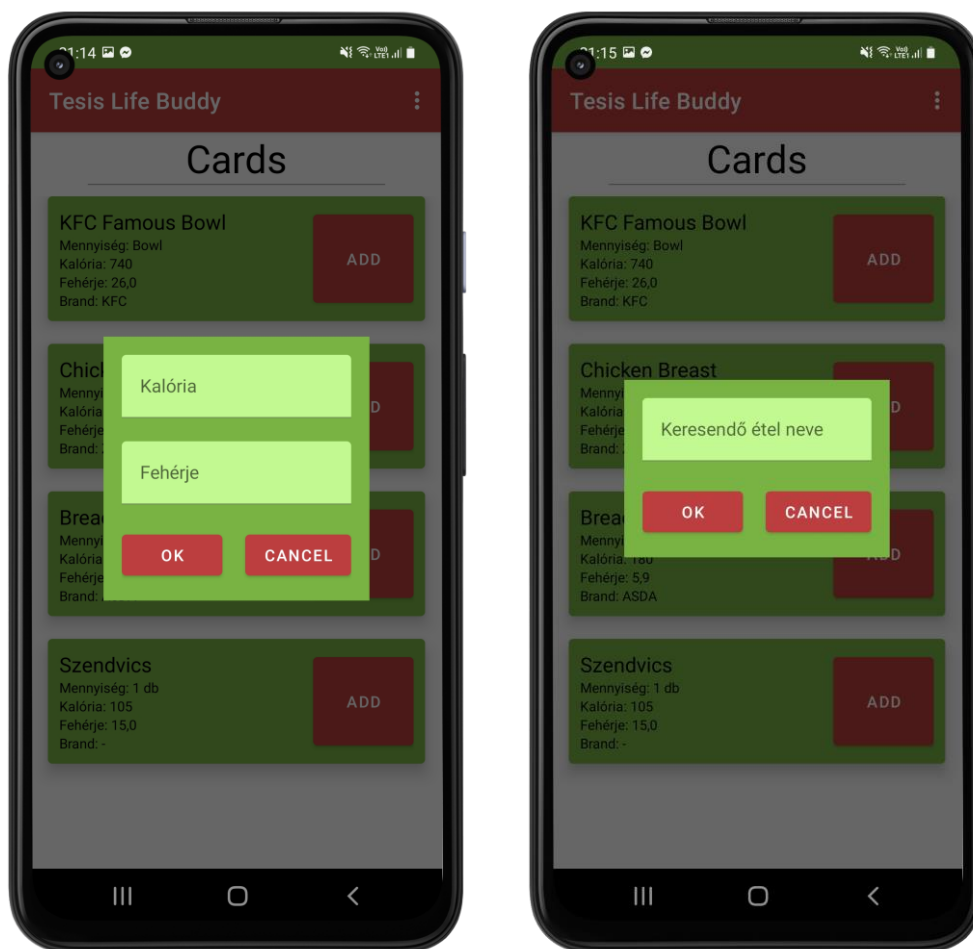
Ha úgy nyomunk az "Ok" gombra, hogy valamelyik kötelező mező üres, akkor az app hibát jelez, egyébként a dialógus bezáródik és az új kártya felkerül a listára. Ha a "Cancel" gombra nyomunk, akkor a dialógus bezáródik, más nem történik.



4. Kép: Bal oldalon a Cards képemű menüje látható, jobb oldalon pedig az "Új kártya" menüpont dialógusablaka

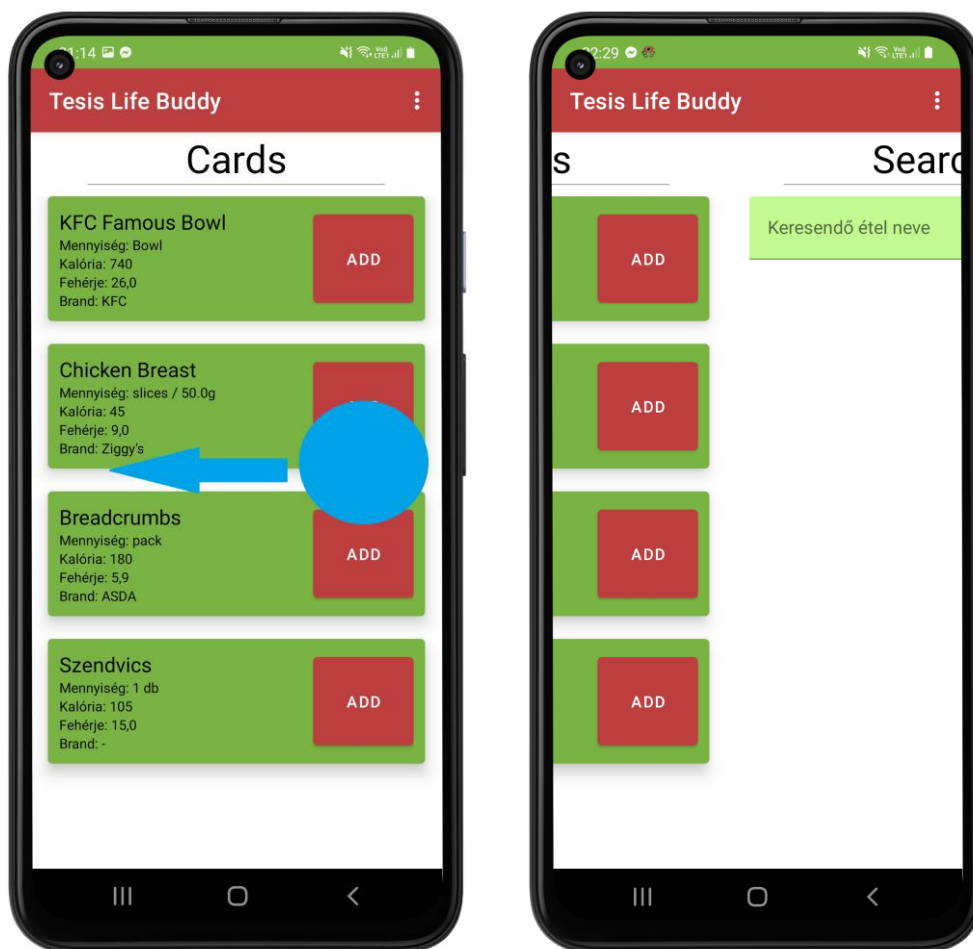
2) "Számlálók növelése": Erre a menüpontra nyomva lehetősége van a felhasználónak növelni a számlálók értékét, anélkül, hogy valamelyik kártyát használná. A megjelenő dialógus ablakban meg kell adni a kalória- és fehérjeértékeket, majd az "Ok" gombra nyomni. A korrábbiakhoz hasonlóan, ha valamelyik mező üres az "Ok" gomb nyomására hibát jelez az app, a "Cancel" megnyomására a dialógus bezáródik és más nem történik.

3) "Keresés": Erre a menüpontra nyomva lehetőség van keresni a kártyák között. A megjelenő dialógus ablakban meg kell adni a keresendő kártya nevét, majd az "Ok" gombra nyomni. Ekkor azok a kártyák, amelyek nevei tartalmazzák a keresendő szöveget, a lista tetejére kerülnek. Ha a mező üres az "Ok" gomb nyomására hibát jelez az app, a "Cancel" megnyomására a dialógus bezáródik és más nem történik.



5. Kép: Bal oldalon a “Szamlalok novelése” menüpont dialógusablaka, jobb oldalon a “Keresés” menüpont dialógusablaka látható

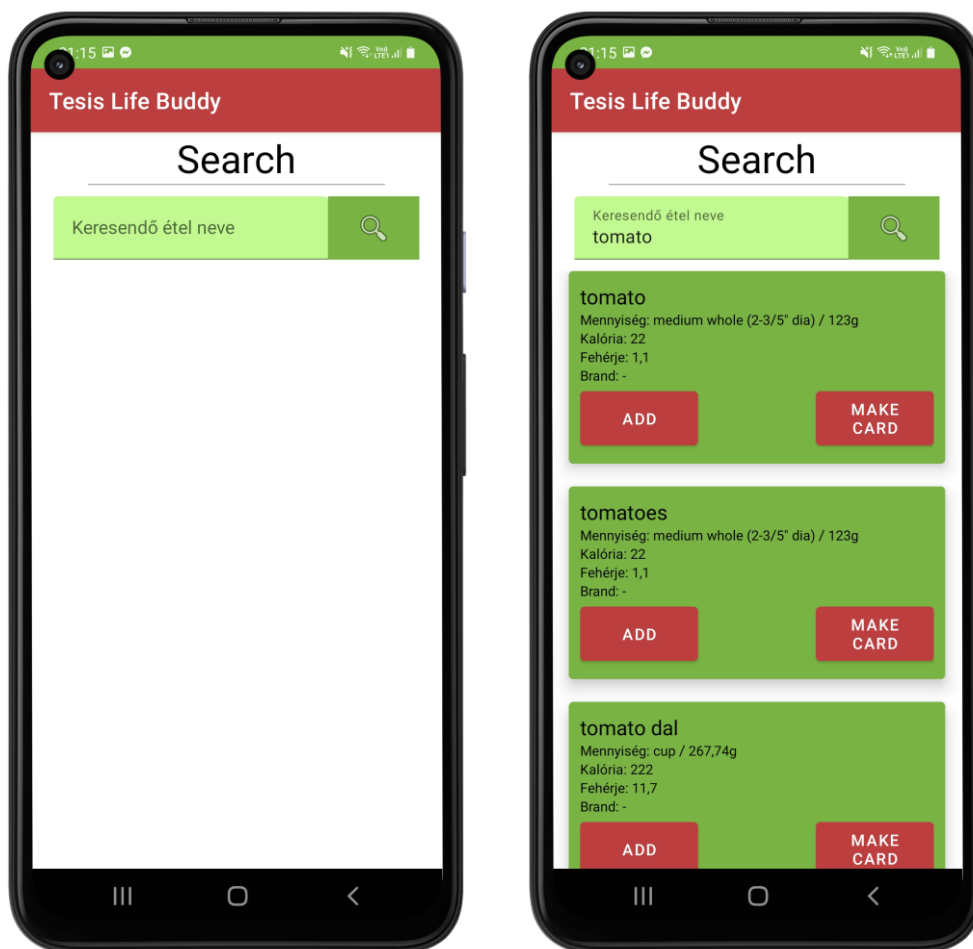
Ha a Cards képernyőn jobbról balra húzzuk az ujjunkat, akkor átlépünk a Search képernyőre. Hasonlóképpen a Search képernyőről úgy tudunk visszalépni a Cards képernyőre, hogy balról jobbra húzzuk az ujjunkat.



6. Kép: A baloldalon a gesztúra van ábrázolva, amit tenni kell, hogy képernyőt váltsunk, a jobb oldalon pedig a képernyőváltás folyamata

A Search képernyőn egy searchbar látható. Ebbe, ha beírjuk a keresett étel (vagy akár brand-jét) és rányomunk a nagyító ikonra, akkor az app kilistázza a Nutritionix adatbázisában talált találatokat. A kapott lista elemein ugyanazok az attribútumok láthatóak, mint a Cards képernyő kártyáin. Található még rajtuk továbbá két gomb: Az "Add" gomb hatása ekvivalens a Cards képernyőn látott "Add"-dal, a "Make Card" gomb hatására az adott elem elmentődik a kártyák közé.

Megjegyzés: Ha az adott elem név, mennyiség vagy brand attribútuma túl hosszú, akkor a "Make Card" gomb hatására nem lesz automatikusan mentve a kártyák közé, ekkor a felhasználónak manuálisan kell azt elmentenie (Cards képernyő "Új kártya" menüpont). Ha ez az eset (túl hosszú attribútumok) fennáll, akkor erre az alkalmazás fel is hívja a felhasználó figyelmét egy snackbar segítségével.



7. Kép: Bal oldalon a Search képernyő látható keresés előtt, jobb oldalon pedig keresés után

A Cards és Search képernyőkről a Back button (a képeken a jobb alsó sarokban látható visszanyíl) segítségével léphetünk vissza a kezdőképernyőre.

Felhasznált technológiák

- A progressbar-ok **CustomView**-ként vannak megvalósítva
- Az alkalmazás **RecyclerView**-t használ a listák megjelenítésére és kezelésére
- Az alkalmazásnak van egy perzisztens **Notification**-je, aminek a megjelenítését egy **Service** végzi
- A Cards és Search képernyők a **Fragment** és **ViewPager** technológiák felhasználásával kerülnek megjelenítésre
- A dialógusok is a **Fragment**-ként kerülnek megjelenítésre
- A kalória- és fehérjeértékek **SharedPreferences**-ben tárolódnak el
- A mentett ételek **SQLite** adatbázisban tárolódnak, az SQLite kényelmes használatához a [Room](#) könyvtárat használtam
- Az alkalmazás képes **hálózati kommunikációra**: a [Nutritionix](#) adatbázisából tudja lekérdezni az ételeket, a http kérések kényelmes lebonyolításához a [Retrofit](#), json adatok konvertálására pedig a [Moshi](#) könyvtárat használtam
- A dialógusok, illetve a listák elemei **animációkkal** jelennek meg

Fontosabb technológiai megoldások

Egyik első probléma, amibe ütköztem, az a kalória- és fehérjeértékek elmentése volt. Ezek egyszerű számadatok, úgyhogy nem tűnt célszerűnek egy SQLite táblát szentelni nekik. Így jött ötletként, hogy használjak SharedPreferences-t, mivel az tökéletesen ellátja az ilyen primitív adatok tárolásának funkcióját. Továbbá így a FoodActivity-nek nem kell, hogy direkt referenciája legyen a MainActivity-re a kalória- és fehérjeértékek változtatásához, hiszen elég neki a megfelelő SharedPreferences értékeket változtatni, amiket a MainActivity a progressbar-ok megjelenítésekor betölt. Így az Activity-k közti kommunikációnak is egy hatékony módja lett.

Ezentúl probléma volt, hogy az SQLite adatbázist mindig FoodActivity kérte le az onCreate metódusában. Egyrészt kényelmetlen volt, mert az Activity-n található Fragment-eknek interfészekon keresztül kellett kommunikálniuk az Activity-vel, ha valamilyen adatbázissal kapcsolatos műveletet kellett végezniük. Másrészt az adatbázis többszöri lekérése (minden onCreate-ben) végett, a cachelések következtében, már elavult állapotok is megjelentek az adatok között (pl. egy már törölt elem szerepelt a listában). Ezt úgy oldottam meg, hogy csináltam egy saját Application osztályt, amelynek az onCreate-jében kértem le az adatbázist, és elmentettem egy globális változóba. Ezáltal csak egyszer van lekérve (az alkalmazás indulásakor), így már nem voltak jelen a hibás állapotok, valamint a globális változón keresztül a Fragment-ek is képesek kényelmesen elvégezni a szükséges műveleteket.