Szegedi Tudományegyetem

Informatikai Intézet

SZAKDOLGOZAT

Tóth Bagi Bence

2022

Szegedi Tudományegyetem

Informatikai Intézet

Kalóriaszámláló fitness alkalmazás

Szakdolgozat

|  |  |  |
| --- | --- | --- |
|  | Készítette: |  |
|  | Tóth Bagi Bence |  |
|  | programtervező informatikus BSc szakos hallgató |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Belső témavezető: |  | Külső témavezető: |  |
|  | Dr. Bilicki Vilmos |  | Jánki Zoltán Richárd |  |
|  | egyetemi adjunktus |  | tudományos segédmunkatárs |  |

Szeged

2022

## Feladatkiírás

A szakdolgozatom során egy olyan mobilalkalmazást kellett megvalósítanom, ami segítségével a felhasználó rögzíteni tudja a napi étkezéseire vonatkozó adatokat, és nyomon tudja követni ezeket. Fontos szempont volt, hogy találjak egy ingyenesen elérhető és szabadon felhasználható adatbázist, ami tartalmazza minél több élelmiszernek a makrotápanyag értékeit. Végezetül meg kellett valósítanom, hogy az adatbázisban rá tudjunk keresni egy konkrét élelmiszerre, a neve alapján.

## Tartalmi összefoglaló

* A téma megnevezése:

Kalóriaszámláló fitness alkalmazás

* A megadott feladat megfogalmazása:

A feladat egy olyan mobilalkalmazás elkészítése, ami képes a felhasználó napi étkezéseire vonatkozó adatokat rögzíteni, és nyomon követni. Miután a felhasználó beregisztrál, és megadja az adatait (nem, életkor, testsúly, magasság, aktivitási szint), valamint a célját (fogyás, jelenlegi testsúly megtartása, tömegnövelés), az alkalmazás meghatároz neki egy személyre szabott, ajánlott napi kalória beviteli értéket. Ez az érték egy tudományosan elismert képlet alapján kerül kiszámításra. Étkezés bevitelénél meg tudjuk kézileg adni a makrotápanyagok értékeit, vagy van lehetőségünk rákeresni egy konkrét élelmiszerre, illetve az elmentett receptjeink közül is tudunk választani.

* A megoldási mód:

Keresnem kellett egy adatbázist, ami tartalmazza minél több élelmiszer kalória és makrotápanyag értékeit. Miután ez sikerült, az adatbázist átalakítottam olyan formátumra, ami nekem megfelel. Ezután megterveztem az alkalmazás adatmodelljét. Végül megírtam az alkalmazást, implementálva a korábban említett adatbázist és adatmodellt.

* Alkalmazott eszközök, módszerek:

Az alkalmazás Dart programozási nyelven íródott, és a Flutter keretrendszert használja.  
A backendért a Google Firebase a felelős, illetve neki a Firestore adatbázisát alkalmaztam.

* Elért eredmények:

Az eredmény egy modern, letisztult kinézetű, felhasználóbarát mobilalkalmazás. Mivel keresztplatformos, ezért Android és iOS operációs rendszeren is egyaránt működik.

* Kulcsszavak:

Mobilalkalmazás, adatbázis, Flutter, Firebase, Firestore

## Tartalomjegyzék

[Feladatkiírás 3](#_Toc103460229)

[Tartalmi összefoglaló 4](#_Toc103460230)

[Tartalomjegyzék 5](#_Toc103460231)

[Motiváció 7](#_Toc103460232)

[1. Terület áttekintése 7](#_Toc103460233)

[1.1. MyFitnessPal 7](#_Toc103460234)

[1.2. YAZIO Fasting & Food Tracker 8](#_Toc103460235)

[1.3. Lifesum: Healthy Eating & Diet 8](#_Toc103460236)

[1.4. Macros – Calorie Counter & Meal Planner 8](#_Toc103460237)

[2. A kalóriaszámlálásról 9](#_Toc103460238)

[2.1. Mi is az a kalória? 9](#_Toc103460239)

[2.2. Ajánlott napi kalória beviteli érték 9](#_Toc103460240)

[3. Felhasznált technológiák 10](#_Toc103460241)

[3.1. Flutter 11](#_Toc103460242)

[3.2. Firebase 11](#_Toc103460243)

[3.3. Node.js 11](#_Toc103460244)

[4. Funkcionális specifikáció 12](#_Toc103460245)

[4.1. Az alkalmazás kezelése 12](#_Toc103460246)

[4.1.1. Használati összefoglaló 12](#_Toc103460247)

[4.1.2. Felhasználó autentikáció 13](#_Toc103460248)

[4.1.3. Adatok megadása 15](#_Toc103460249)

[4.1.4. Kezdőlap 16](#_Toc103460250)

[4.1.5. Étkezés hozzáadása 16](#_Toc103460251)

[4.1.6. Drawer 18](#_Toc103460252)

[4.1.7. Napló oldal 19](#_Toc103460253)

[4.1.8. Receptjeim oldal 20](#_Toc103460254)

[4.1.9. Adataim oldal 21](#_Toc103460255)

[5. Architektúra 22](#_Toc103460256)

[5.1. Flutter projekt struktúrája 22](#_Toc103460257)

[5.1.1. Models 22](#_Toc103460258)

[5.1.2. Screens 24](#_Toc103460259)

[5.1.3. Widgets 25](#_Toc103460260)

[5.1.4. Main.dart 25](#_Toc103460261)

[5.2. Felhasznált csomagok 25](#_Toc103460262)

[6. Adatmodell 27](#_Toc103460263)

[6.1. Foods kollekció 27](#_Toc103460264)

[6.2. Users kollekció 27](#_Toc103460265)

[6.2.1. Meals kollekció 28](#_Toc103460266)

[6.2.2. Recipes kollekció 28](#_Toc103460267)

[7. Az étel adatbázis 29](#_Toc103460268)

[7.1. Keresőmező hozzáadása az adatbázishoz 30](#_Toc103460269)

[7.2. JSON fájl feltöltése a Firestore-ra 30](#_Toc103460270)

[Irodalomjegyzék 31](#_Toc103460271)

[Nyilatkozat 32](#_Toc103460272)

[Mellékletek 33](#_Toc103460273)

## Motiváció

Több tanulmány is bebizonyította már, hogy azok a személyek, akik feljegyzik és nyomon követik napi étkezéseiket és fizikai aktivitásukat, nagyobb eséllyel érik el fitnesz céljaikat. Minél rendszeresebben teszik ezt, annál jobb lesz a végeredmény. Alkalmazásom segíteni tud abban, hogy az emberek könnyen, gyorsan, és precízen fel tudják jegyezni, és nyomon tudják követni az étkezéseikre vonatkozó adataikat.

Célom, hogy megvalósítsak egy modern, felhasználóbarát mobilalkalmazást, amivelsegítséget tudok nyújtani azoknak az embereknek, akik egészségesebben szeretnének táplálkozni, és javítani szeretnének a külső megjelenésükön.

## Terület áttekintése

A Google Play áruházban, és az Apple App Store-ban is nagyon sok fajta fitneszes és edzős alkalmazás található. Szerintem 3 fő tényező van, ami miatt egyikük sem tökéletes:

* Nem felhasználóbarátok

Több alkalmazással is találkoztam, aminek a kezelése és felhasználói felülete véleményem szerint túl összetett volt. Több funkciót is tartalmaznak, amikre az átlagos felhasználónak nincs szüksége, illetve nem egyértelmű számukra, hogy mi mit jelent és mit csinál. Ebből következik, hogy az alkalmazás zsúfolt és nehezen kezelhető lesz, ami negatív kihatással van a felhasználói élményre.

* Egyes funkciók csak a fizetett verzióban érhetőek el

Jelenleg, az áruházban található appok nagy része rendelkezik fizetett változattal.

Persze ettől még lehet ingyenesen is használni az alkalmazást, viszont nagy eséllyel valamilyen módon le leszünk korlátozva. Sok esetben olyan funkciókkal hirdetik az alkalmazást, vagy az alkalmazás leírásánál olyan funkciók szerepelnek, amiket csak a fizetett verzióban lehet használni – és ezt csak azután tudjuk meg, miután már letöltöttük az alkalmazást, és beregisztráltunk, valamit megadtuk nekik személyes adatainkat.

* Nem támogatják a magyar nyelvet

## MyFitnessPal

A MyFitnessPal a jelenlegi legnépszerűbb fitnesz alkalmazás. A Play áruházban levő leírása szerint 200 millió felhasználója van, és hozzáférést nyújt a világ legnagyobb élelmiszer adatbázisához, ami tartalmazza több mint 14 millió élelmiszernek a pontos makrotápanyag értékeit.

Az alkalmazás hátrányai: nem felhasználóbarát, és nem használható magyar nyelven. Az egyik alap funkció, hogy egy étkezésnél egyedi makrotápanyag értékeket tudjunk megadni, csak a fizetett verzióban érhető el.

## YAZIO Fasting & Food Tracker

A YAZIO alkalmazás nekem személy szerint jobban tetszik, mint a MyFitnessPal. Letisztultabb a felhasználói felülete, könnyebb használni, és használható magyar nyelven is. Ebben az app-ban már előfizetés nélkül is tudunk étkezés bevitelnél egyedi makrotápanyag értékeket megadni. Hátránya viszont, hogy nem tudunk saját recepteket létrehozni, ahol mi adjuk meg a hozzávalókat, hanem csak az alkalmazásban találhatókat tudjuk használni. Másik nagy hátránya, hogy előfizetés nélkül a nyomon követett adatainkat csak 30 napra visszamenőleg tudjuk megtekinteni.

## Lifesum: Healthy Eating & Diet

A Lifesum több részben is hasonlít a YAZIO-ra. Letisztult, modern kinézetű, egyszerű kezelni. Mindent tud, amit egy ilyen alkalmazásnak tudnia kellene. Hátránya, hogy az ingyenes verzióban nem tudunk saját recepteket létrehozni, csak ha előfizetünk, illetve nem támogatja a magyar nyelvet.

## Macros – Calorie Counter & Meal Planner

A Macros egy nagyon egyszerű alkalmazás, használata könnyű és egyértelmű, nem tartalmaz extra, felesleges funkciókat. Minden fő funkció, ami egy átlagos felhasználó számára szükséges, megtalálható benne. Hátránya, hogy nem használható magyar nyelven.

## A kalóriaszámlálásról

A kalóriaszámlálás fogalma azt jelenti, hogy minden étkezésünknek kiszámoljuk a kalória és makrotápanyag értékeit, majd ezeket feljegyezzük és nyomon kövessük. Ennek előnye, hogy pontosan tudni fogjuk, hogy mennyi kalóriát viszünk be a szervezetünkbe, és ennek segítségével megfelelően állíthatjuk be a céljainkat. Több tanulmány is bebizonyította már, hogy azok a személyek, akik számolják kalóriáikat, nagyobb eséllyel érik el fitnesz céljaikat.

* 1. Mi is az a kalória?

A kalória (cal) az energia mértékegysége. A kalóriát Nicolas Clément vezette be a hő mérésére 1824-ben. Az SI-mértékegységrendszer bevezetésével a joule nagyrészt kiváltotta, ma főleg az élelmiszerek energiatartalmának mérésére használják. Ez a meghatározás valójában az ú.n. kiskalória, más néven grammkalória definíciója.  
A táplálkozástudomány és a hétköznapi élet egy ennél ezerszer nagyobb mértékegységet nevez kalóriának, aminek a neve kilokalória (kcal).

Minden élelmiszer rendelkezik makrotápanyagokkal (fehérje, szénhidrát, zsír). Ha egy adott elélmiszernek tudjuk a makrotápanyag értékeit, akkor ki tudjuk számolni, hogy hány kalória van benne. A szénhidrát és a fehérje 4 kalóriát tartalmaznak grammonként, a zsír pedig 9-et. Tehát egy élelmiszer kalória értékét az alábbi képlettel tudjuk kiszámolni (a kalóriát jelölje *k* betű, a fehérjét *f*, a szénhidrátot *s*, és a zsírt *z*).  
A kalóriát kcal-ban mérjük a makrotápanyagok értékeit grammban.

*k* = *f* · 4 + *s* · 4 + *z* · 9   
*(2.1)*

* 1. Ajánlott napi kalória beviteli érték

Az ajánlott napi kalória beviteli érték meghatározza, hogy egy személynek hány kalóriát kell bevinnie a szervezetébe naponta, hogy elérje célját, legyen az fogyás, tömegnövelés, vagy a jelenlegi testsúly megtartása. Alkalmazásomban a Mifflin St. Jeor egyenletet alkalmaztam a napi kalória beviteli értlék kiszámítására. Ennek az értéknek a kiszámolása több lépésből áll. Először is ki kell számolnunk az alapanyagcsere-értékünket. Ez az érték azt a kalóriamennyiséget jelöli, amelyet a szervezet az alap életfenntartó funkciók ellátása során eléget egy nap folyamán. Ez az érték különbözik a férfiak és nők esetében.  
Az alapanyagcsere-értéket jelölje a *b* betű, a testsúlyt *w*, a magasságot *h*, és az életkort *a*. Az alapanyagcsere- értéket a következő képlettel tudjuk kiszámolni:

* férfiak esetében:

*b* = *w* · 10 + *h* · 6.25 - *a* · 5 + 5   
*(2.2)*

* nők esetében:

*b* = *w* · 10 + *h* · 6.25 - *a* · 5 - 161  
*(2.3)*

Az alapanyagcsere-értéket kcal-ban mérjük, a testsúlyt kilogrammban, a magasságot centiméterben, és az életkort pedig években. Miután ezt megkaptuk, ezt az értéket be kell szoroznunk az aktivitási szintünkkel, ami a következőképpen alakul ki:

* Mozgásszegény: 1.2
* Enyhén aktív: 1.375
* Közepesen aktív: 1.550
* Nagyon aktív: 1.725
* Extra aktív: 1.9

A kapott kalória értéket ez után módosítanunk kell a célunk alapján: fogyás esetén csökkentjük 15 százalékkal, tömegnövelés esetén hozzáadunk 500 kcal-t, és jelenlegi testsúly megtartása esetén pedig változatlan marad. Így kapjuk meg végül az ajánlott napi kalória beviteli értéket. Ezt a kalóriaszámot a következő arányokban osztjuk fel makrotápanyagokra, a célunktól függően:

* fogyás: 40/40/20 arányban (szénhidrát/fehérje/zsír)
* tömegnövelés: 40/30/30
* súlyfenntartás: 40/30/30

Látható, hogy a tömegnövelés és súlyfenntartás esetén megegyeznek az arányok, viszont a kalóriák és makrotápanyagok értékei eltérnek.

## Felhasznált technológiák

Ebben a részben a szoftver fejlesztése során alkalmazott technológiákat mutatom be röviden.

* 1. Flutter

Az alkalmazásomat a Flutter keretrendszer segítségével valósítottam meg. A Flutter az a Google által fejlesztett és fenntartott, nyílt forráskódú keretrendszer, amivel natívan fordított keresztplatformos alkalmazásokat tudunk fejleszteni. Nagy előnye, hogy egyetlen kódbázisból képes elkészíteni ugyanazt az alkalmazást akár Androidra, iOS-re, Windowsra, Linuxra, MacOS-re, Google Fuchsia-ra, vagy akár a webre is. Ezek közül nekünk az Android és az iOS az érdekes, mivel azok mobil operációs rendszerek. A Flutter a Dart programozási nyelvet használja, amit szintén a Google fejlesztett ki 2011-ben. A Dart az főként kliens oldali fejlesztésre lett kitalálva, mint például a mobil appok és a web. Objektum orientált nyelv, opcionálisan típusos, C-stílusú szintaxist használ, valamint képes natív gépi kódra, illetve JavaScriptre fordulni.

* 1. Firebase

A Firebase az egy Google által fejlesztett felhőalapú platform, ami backend szolgáltatásokat nyújt mobilos és webes alkalmazások számára. Fő szolgáltatásai: valós idejű adatbázis, felhőalapú tárhely, felhasználó autentikáció, összeomlás jelentés, gépi tanulás, távoli konfigurálás, és statikus állományok tárolása. Ezek a szolgáltatások közül én az autentikációt, és a Firestore adatbázist vettem igénybe.

A Firebase Authentication biztosít egy biztonságos és jól működő autentikációs szolgáltatást. A felhasználók tudnak regisztrálni és bejelentkezni: email cím és jelszó párossal, telefonszámmal, vendégként, illetve külső szolgáltatók fiókaival is, mint például: Google, Facebook, Apple, stb… fiókokkal. Könnyen integrálható különböző alkalmazásokba a Firebase Authentication SDK-nak köszönhetően.

A Firestore egy NoSQL adatbázis, ami a kollekció-dokumentum adatmodellt követi. A kollekció az egy táblaként szolgál, a dokumentumok pedig a rekordok. A dokumentumok tartalmazhatnak alkollekciókat is. Felhőalapú adatbázis, ami adatok tárolására és szinkronizálására szolgál. Gyors, flexibilis és skálázható, valamint alkalmazható kliens- és szerver-oldali fejlesztéshez is.

* 1. Node.js

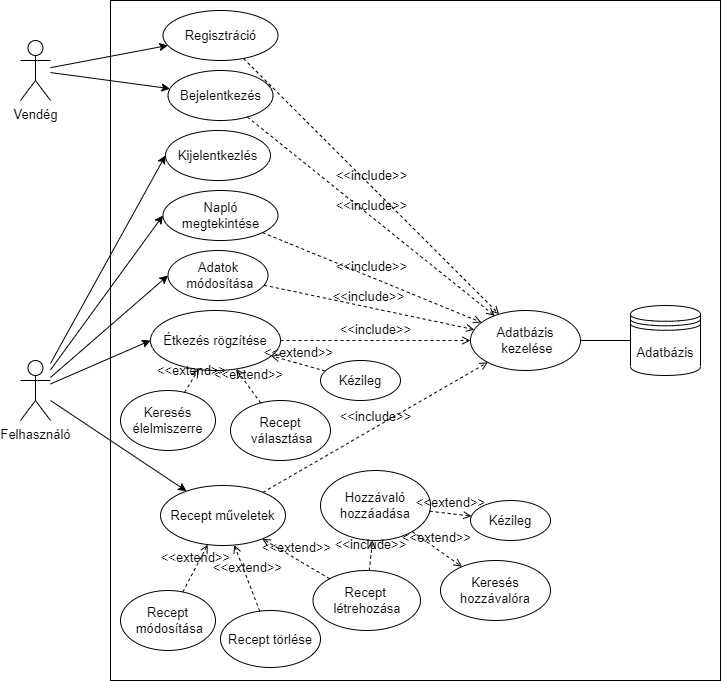
A Node.js egy nyílt forráskódu, keresztplatformos, szerveroldali JavaScript futtatókörnyezet, ami a Google-féle V8 motoron fut. Képes JavaScript kódot végrehajtani, böngésző használata nélkül. A fejlesztés során alkalmaztam egy Node.js szkriptet, ami lehetővé tette, hogy egy JSON fájlt fel tudjak tölteni a Firestore-ra úgy, hogy abból egy kollekció keletkezzen. Ennek működését és folyamatát majd a későbbiekben mutatom be.

## Funkcionális specifikáció

Alkalmazásom fő funkciója, hogy lehetővé tegye a felhasználóknak, hogy rögzíteni tudják a napi étkezéseikre vonatkozó adatokat, és nyomon tudják követni ezeket. Az alkalmazás használata előtt a felhasználó be kell, hogy jelentkezzen, illetve, ha még nem rendelkezik fiókkal, akkor regisztrálnia kell. Az autentikációs műveleteket a Firebase Authentication végzi el. Sikeres regisztráció után meg kell adnunk a személyes adatainkat (nem, életkor, testsúly, magasság, aktivitási szint), valamint a célunkat (fogyás, jelenlegi testsúly megtartása, tömegnövelés). Ez után az alkalmazás meghatároz nekünk egy személyre szabott, ajánlott napi kalória beviteli értéket. Ez az érték egy tudományosan elismert képlet alapján kerül kiszámításra. Szükség esetén bármikor változtatni tudunk adatainkon. Az adatok módosítása után az ajánlott kalória érték automatikusan újra kiszámítódik. Amikor egy új étkezést szeretnénk rögzíteni, lehetőségünk van kézileg megadni a makrotápanyagok értékeit, illetve rá tudunk keresni egy adatbázisból egy konkrét élelmiszerre a neve alapján, vagy az elmentett receptjeink közül is tudunk választani. Adatbázisként egy nyilvánosan elérhető és szabadon felhasználható adatbázist használtam fel, ami tartalmazza a kalória és makrotápanyag értékeit több mint 5700 élelmiszernek. Ez az adatbázis több módosításon és átalakításon is átesett, mire felkerült a Firebase Firestore-ra. Ennek a folyamatáról majd a későbbiekben fogok részletesen beszámolni.

* 1. Az alkalmazás kezelése
     1. Használati összefoglaló

Az alkalmazás használata során a felhasználó sok féle műveletet hajthat végre, melyeket érdemes összesíteni és ábrázolni. Erre a célra szolgál a use case diagram (használati eset diagram). A use case diagram a felhasználó és a rendszer lehetséges interakcióinak grafikus ábrázolása. Különböző használati eseteket, és a rendszer különböző típusú felhasználóit mutatja be. Az alkalmazás use case diagramja a 4.1. ábrán látható.



4.1. ábra – Az alkalmazás use case diagramja

Az ábrán látható, hogy a felhasználó milyen műveleteket tud végrehajtani az alkalmazás használata során. Fel van tüntetve, hogy melyik műveletek létesítenek kapcsolatot az adatbázissal. Mivel a műveletek túlnyomó többségének szüksége van az adatbázishoz való hozzáféréshez, ezért elengedhetetlen az alkalmazásnak a folyamatos internetkapcsolat, mivel az adatbázis a felhőben helyezkedik el, és nem lokálisan a kliens oldalon. A műveletek közül csak a kijelentkezés végezhető el internetkapcsolat nélkül, mivel az az eszköz lokális tárából törli a bejelentkezéssel kapcsolatos adatokat.

* + 1. Felhasználó autentikáció

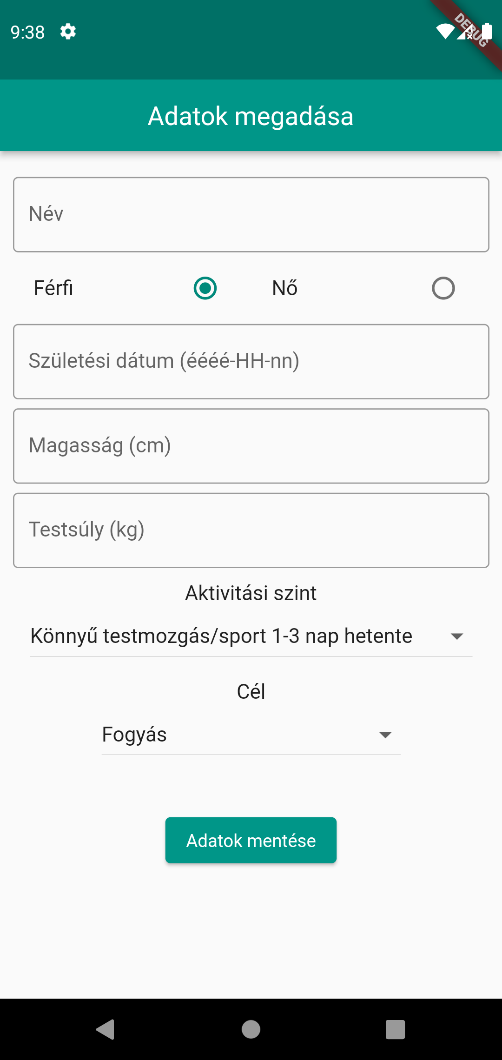
Az alkalmazás fontos részét képezi a felhasználók kezelése, mivel azt szeretnénk elérni, hogy minél többen használják az appot, és értelemszerűen minden felhasználó eltérő adatokkal rendelkezik.

Az alkalmazást elindítva, ha nem jelentkeztünk be korábban, az autentikációs képernyő (4.2. ábra) fogad bennünket.

|  |  |
| --- | --- |
| A képen szöveg látható  Automatikusan generált leírás  4.2. ábra – Autentikációs képernyő, bejelentkezés (magyar nyelv) | 4.3. ábra – Autentikációs képernyő, regisztráció (angol nyelv) |

Az alkalmazás képes észlelni a mobileszközön beállított rendszer nyelvet. Első indításnál, ha a rendszer nyelve magyar, akkor az alkalmazás magyarul indul, ha pedig valami más, akkor angolul. Ha már rendelkezünk fiókkal, akkor írjuk be email címünket és jelszavunkat, majd a felső „BEJELENTKEZÉS” gombra koppintva be tudunk jelentkezni. Sikeres bejelentkezés esetén át leszünk navigálva a kezdőlapra. Az alsó, zöld betűszínű gombbal tudunk váltani a bejelentkezés és regisztráció nézetek között. Ha a regisztrációs nézet (4.3. ábra) van kiválasztva, meg tudjuk adni email címünket és jelszavunkat, valamint a jelszót újra be kell írni, a gépelési hibák elkerülése érdekében. Egy email címhez csak egy regisztrált felhasználó tartozhat, és az email cím formailag helyes kell, hogy legyen. A jelszó legalább 6 karakterből kell, hogy álljon. Ha a felső „REGISZTRÁCIÓ” gombra koppintunk, akkor először a kliens validálja az email cím formai helyességét és a jelszó erősségét, és ha ez sikeres, csak akkor küldi el a regisztrációs kérelmet a szerver felé. Ha a Firebase Authentication sikeresen létre tudta hozni fiókunkat, akkor az alkalmazás át irányít minket a következő képernyőre, ahol rögzíteni tudjuk személyes adatainkat (4.4. ábra).

* + 1. Adatok megadása



4.4. ábra – Adatok megadása

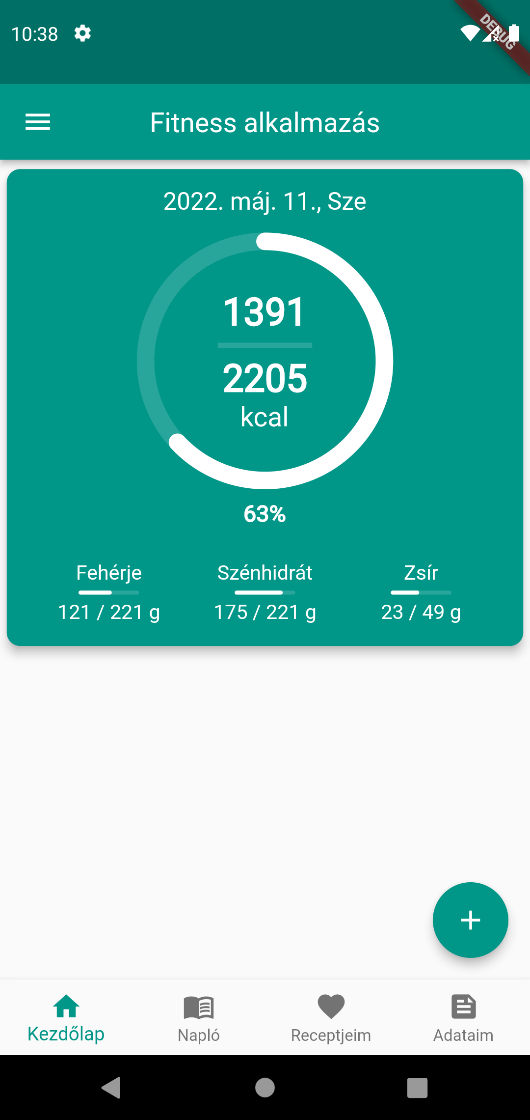
Ez a képernyő arra szolgál, hogy a felhasználó be tudja vinni személyes adatait: név, nem, születési dátum, magasság, testsúly, aktivitási szint, cél. A választható aktivitási szintek a következők:

* Alacsony (kevés, vagy semennyi testmozgás)
* Könnyű testmozgás/sport 1-3 nap hetente
* Közepes testmozgás/sport 3-5 nap hetente
* Nehéz testmozgás/sport 6-7 nap hetente
* Extrém testmozgás/sport 6-7 nap hetente

Három fajta cél közül tudunk választani: fogyás, tömegnövelés, jelenlegi testsúly megtartása. A néven kívül azért kell megadnunk ezeket az adatokat, hogy az alkalmazás ki tudja számolni nekünk az ajánlott napi kalória beviteli értékünket. Ez a képernyő, és az alkalmazás többi része is a metrikus mértékegységrendszert használja. Az „Adatok mentése” gombra koppintva, az autentikációs képernyőhöz hasonlóan, itt is a kliens előbb validálja a beviteli mezőket, mielőtt elküldi az adatokat a szerverre. Ha minden rendben van, akkor az adatok felkerülnek a Firestore-ra, és az alkalmazás át navigál minket a kezdőlapra.

* + 1. Kezdőlap

Ha már korábban bejelentkeztünk, és nem jelentkeztünk ki, akkor az alkalmazás megnyitása után a kezdőlapra kerülünk (4.5. ábra).



4.5. ábra – Kezdőlap

A kezdőlapon fel van tüntetve, hogy hány kalóriát vittünk be eddig a mai napon, és hogy mennyi az ajánlott. Ezek aránya egy kör alakú folyamatjelző sávval van ábrázolva. Ez alatt a bevitt makrotápanyagok mennyisége látható, vízszintes folyamatjelző sávokkal ábrázolva. A képernyő jobb alsó sarkában található hozzáadás gombra koppintva tudunk új étkezést rögzíteni. A képernyő legalján a navigációs sáv helyezkedik el, ami a négy fő képernyő közötti navigációt valósítja meg. Az oldalak között kétféleképpen tudunk lépkedni: a navigációs sávban levő ikonokra koppintva, illetve az ujjunk jobbra vagy balra húzásával a képernyőn. Az alkalmazás négy fő képernyője: kezdőlap, napló, receptjeim, adataim.

* + 1. Étkezés hozzáadása

A kezdőlapról a hozzáadás gomb koppintásával jutunk az étkes hozzáadása képernyőre (4.6. ábra). Ez az oldal külön nyílik meg, nem tudunk róla a többi oldalra navigálni.   
A fejléc bal szélén lévő vissza gomb koppintásával tudunk visszatérni a kezdőlapra. Értelemszerűen ez az oldal arra szolgál, hogy egy új étkezést tudjunk rögzíteni.

A képen szöveg látható

Automatikusan generált leírás

4.6. ábra – Étkezés hozzáadása

Étkezés hozzáadásánál meg kell adni az étkezés jellegét, az étkezéshez tartozó makrotápanyagok értékeit, valamint az elfogyasztott étel nevét. Az étkezés jellege lehet: reggeli, ebéd, vacsora, uzsonna. Az étel makrotápanyag értékeit háromféleképpen tudjuk megadni:

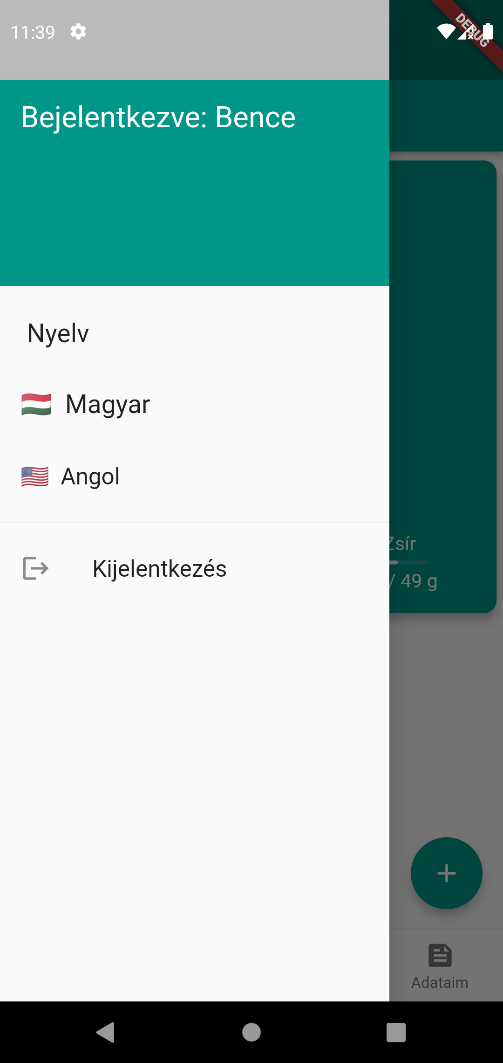
* kézileg be tudjuk gépelni egyenként az értékeket
* rá tudunk keresni az adatbázisban egy adott ételre, majd választani tudunk a találatok közül
* elmentett receptjeink közül tudunk választani

|  |  |
| --- | --- |
| 4.7. ábra – „Lazac” ételre való keresés | A képen asztal látható  Automatikusan generált leírás  4.8. ábra – Recept kiválasztása |

A 4.7. ábrán látszik, hogy a „Keresés ételre” mezőbe beírtuk azt, hogy lazac, majd a fejléc jobb szélén lévő keresés gombra koppintva megjelent egy felugró ablak, amiben ki vannak listázva a találatok. Először be kell írni a kívánt mennyiséget, majd ha rákoppintunk valamelyik találatra, akkor az ablak eltűnik, és a makrotápanyagok beviteli mezői automatikusan kitöltődnek a megfelelő értékekkel. Ha a keresés melleti „recept könyv” gombra koppintunk, akkor egy felugró ablakban megjelennek az elmentett receptjeink (4.8. ábra). Hasonlóan a kereséshez, ha rákoppintunk valamelyik elemre, akkor az ablak eltűnik, és a mezők kitöltődnek. A 4.6. ábrán látható „Hozzáadás” gomb megnyomásával elmentődik az étkezés, és felkerül a Firestore adatbázisba.

* + 1. Drawer

Az alkalmazás rendelkezik egy drawer-el (4.9. ábra), ami az összes fő képernyőről megnyitható, a fejléc bal szélén elhelyezkedő ikonra koppintva.



4.9. ábra – Drawer

A drawer fejlécében ki van íratva a jelenleg bejelentkezett felhasználó neve. Törzsében ki tudjuk választani, hogy az alkalmazás a magyar, vagy az angol nyelvet használja, illetve itt található a kijelentkezés gomb. A kijelentkezésre koppintva a felhasználó ki lesz jelentkeztetve, és át lesz navigálva az autentikációs képernyőre.

* + 1. Napló oldal

A napló oldal (4.10. ábra) a navigációs sávban a második helyen helyezkedik el. Ez az oldal arra szolgál, hogy a felhasználó vissza tudja nézni, és nyomon tudja követni korábban bevitt étkezéseit. Információt kapunk a bevitt kalóriákról és makrotápanyagokról, valamint, az étkezések jellegéről és az elfogyasztott étel nevéről. Amikor először navigálunk a napló oldalra, akkor az aktuális napnak az adatai lesznek betöltve. A fenti nyilakra koppintva tudunk egyesével lépkedni a napok között. Ha rá koppintunk maga a dátumra, akkor megjelenik egy naptár, ami segítségével ki tudunk választani egy tetszőleges dátumot (4.11. ábra).

|  |  |
| --- | --- |
| 4.10. ábra – Napló oldal | 4.11. ábra – Dátum kiválasztása naptárból |

* + 1. Receptjeim oldal

Ez az oldal a receptek a kezelésére szolgál (4.12. ábra). A felhasználó létre tud hozni új receptet, megtekintheti a meglévő receptjeit, valamint módosítani és törölni is tudja őket.



4.12. ábra – Receptjeim oldal

Új recept létrehozásánál az első lépés, hogy nevet adunk neki. Ez után egyesével hozzá kell adnunk a recept hozzávalóit. Egy recept tetszőleges mennyiségű hozzávalót tartalmazhat. Minden hozzávalónak meg kell, hogy adjuk a nevét, mennyiségét, valamint a fehérje, szénhidrát és zsír értékeit (4.13. ábra). Be tudjuk vinni kézileg is az értékeket, vagy a „Hozzávaló neve” mező kitöltésével, majd a keresés ikon koppintásával rá tudunk keresni a kívánt hozzávalóra (az élelmiszerre való keresés ugyanúgy működik, mint az étkezés hozzáadásánál). Ha rá koppintunk a sarokban lévő mentés ikonra, akkor a jelenleg beírt hozzávaló hozzá lesz adva a recepthez, és a beviteli mezőkből a beírt szövegek törlődnek. Ez után lehetőségünk van újabb hozzávaló hozzáadására, vagy ha végeztünk, akkor el tudjuk menteni a receptet. Recept módosításánál át tudjuk nevezni a receptet, tudunk új hozzávalókat hozzáadni, valamint a meglévő hozzávalóit is tudjuk módosítani, vagy törölni (4.14. ábra).

|  |  |
| --- | --- |
| A képen szöveg látható  Automatikusan generált leírás  4.13. ábra – Hozzávaló hozzáadása új recepthez | A képen szöveg látható  Automatikusan generált leírás  4.14. ábra – Megnyitott recept (módosítás) |

* + 1. Adataim oldal

Ezen az oldalon tudja a felhasználó módosítani személyes adatait. Az „Adatok mentése” gombra kattintva elmentődnek a beírt adatok a Firestore-ba, és újra kiszámítódik az ajánlott napi kalória beviteli érték. Az oldal kinézete a 4.15. ábrán látható.

A képen szöveg látható

Automatikusan generált leírás

4.15. ábra – Adataim oldal

## Architektúra

* 1. Flutter projekt struktúrája

A Flutter projektom könyvtár struktúrája (5.1. ábra) kialakításánál arra törekedtem, hogy a felépítése minél egyszerűbb legyen. Ez biztosította számomra, hogy a fejlesztés folyamata könnyen és zökkenőmentesen történjen.

A képen szöveg látható

Automatikusan generált leírás

*5.1. ábra – Projekt könyvtár struktúrája*

Az osztályokat három csomagba szerveztem: models, screens, widgets.

* + 1. Models

A models mappa tartalmazza a projekt entitásait reprezentáló osztályokat (5.2. ábra).

A képen szöveg látható

Automatikusan generált leírás

5.2. ábra – Entitás osztályok

Csupán három entitásunk van:

* meal: egy étkezést reprezentál
* recipe: egy receptet reprezentál
* ingredient: egy receptnek egy hozzávalóját reprezentálja

A képen szöveg, képernyő látható

Automatikusan generált leírás

5.2. ábra – Meal ostzály

Az 5.2. ábrán a meal osztály tulajdonságait figyelhetjük meg a kódrészleten. A következő tulajdonságokat tartalmazza: az étkezés jellege, az étel neve, fehérje-, szénhidrát-, zsír-, kalóriák mennyisége. Látszik, hogy a kalória az a makrotápanyagok értéke alapján számítódik ki.

A képen szöveg látható

Automatikusan generált leírás

5.3. ábra – Ingredient osztály

Az 5.3. ábrán az ingredient osztály tulajdonságai látható. A következő tulajdonságokat tartalmazza: a hozzávaló neve, mennyisége, fehérje, szénhidrát, zsír, kalóriák. Az étkezés osztályhoz hasonlóan, itt is a makrotápanyagok értékei alapján számítódik ki a kalória értéke.

A képen szöveg látható

Automatikusan generált leírás

5.4. ábra – Recipe ostzály

A recipe osztály (5.4. ábra) a recepteket reprezentálja. Tartalmazza a recept nevét, fehérje-, szénhidrát-, zsír-, kalória értékeket, valamint a recept hozzávalóit. A recept hozzávalói a korábbi ingredient osztályból álló listaként vannak tárolva.

* + 1. Screens

A screens mappa tartalmazza az alkalmazás képernyőit (oldalait) megvalósító osztályokat (5.5. ábra). Mindegyik osztály egy-egy képernyőt reprezentál. Ebben az esetben a képernyőre a teljes képernyőket értjük, amik tartalmazzák a Scaffold elemet. Az egyes osztályokban megtalálhatók a képernyők elrendezésére és kinézetére vonatkozó kódok, valamint az összes művelet implementációja, amiket az adott képernyőn végezhetünk.  
Az alkalmazás képernyői a következők: étkezés hozzáadása, autentikációs képernyő, napló, kezdőlap, adataim, receptjeim, adatok megadása.

A képen szöveg látható

Automatikusan generált leírás

5.5. ábra – Screens mappa osztályai

* + 1. Widgets

A widgets mappában egyetlen egy osztályunk van, a main\_controller (5.6. ábra).



5.6. ábra – Widgets mappa

A main\_controller osztályban vannak megvalósítva azok az elemek, amelyek az alkalmazás több képernyőjén is megtalálhatóak. Az elemek elrendezésére is kinézetére, valamint a funkcionalitásukra vonatkozó kód is ebben az osztályban helyezkedik el. Ilyen elemek például a drawer, és a képernyő alján levő navigációs sáv. Korábban említettem, hogy az oldalak között úgy is tudunk lépkedni, hogy jobbra, illetve balra húzzuk az ujjunkat a képernyőn. Ennek a funkcionalitása is itt kerül megvalósításra, mivel az oldalak közötti lépkedés kötődik a navigációs sávhoz. Amikor másik oldalra lépünk, akkor a navigációs sáv frissül, hogy azt az oldalt jelölje, amin épp vagyunk.

* + 1. Main.dart

Minden Flutter projektben kötelező, hogy ott legyen a main.dart fájl. Ebben a fájlban található az egész projektre kiható beállítások és konfigurációk. Az én alkalmazásom esetében ezek a következők:

* kapcsolat létesítése a Firebase-el
* lokalizáció, magyar és angol nyelvek beállítása
* az alkalmazás témájának beállítása
* az alkalmazás route-jainak (útvonalainak) beállítása
  1. Felhasznált csomagok

A Flutter lehetővé teszi, hogy más fejlesztők által létrehozott és megosztott csomagokat használjunk projektünkben. A csomagok használata felgyorsítja az alkalmazás fejlesztését, mivel nem kell mindent a nulláról megírni. Úgy tudunk csomagokat hozzáadni projektünkhöz, hogy a pubspec.yaml fájlban, a dependencies (függőségek) szekcióhoz odaírjuk a csomag nevét, és a használni kívánt verziót. Az én alkalmazásomban felhasznált csomagok listája a 5.7. ábrán látható.

A képen szöveg látható

Automatikusan generált leírás

5.7. ábra – Pubspec.yaml fájl dependencies szekciója

* cupertino\_icons: lehetővé teszi az iOS stílusú ikonok használatát
* cloud\_firestore: a Cloud Firestore API használatához szükséges, ami lehetővé teszi, hogy a Firestore-on különféle műveleteket tudjunk végrehajtani
* firebase\_core: a Firebase Core API használatához szükséges, ami lehetővé teszi a csatlakozást több Firebase alkalmazáshoz is
* firebase\_auth: a Firebase Authentication API használatához szükséges, ami lehetővé teszi a felhasználók autentikációját
* fluttertoast: lehetővé teszi, hogy egyszerűen tudjunk toast üzenetek létrehozni
* uuid: lehetővé teszi a UUID (Universal Unique Identifier) generálását
* google\_fonts: lehetővé teszi, hogy a fonts.google.com oldalon lévő betűtípusokat használjuk alkalmazásunkban
* percent\_indicator: lehetővé teszi vízszintes és kör alakú folyamatjelző sávok használatát
* easy\_localization: lehetővé teszi az alkalmazás többnyelvűségének implementálását
* intl: lokalizációhoz, dátumok formázásához használatos

## Adatmodell

Az adatok tárolására a Firebase-nek a Firestore adatbázisát alkalmaztam. A Firestore egy felhőalapú NoSQL adatbázis, ami a kollekció-dokumentum adatmodellt követi.  
A kollekció az egy táblaként szolgál, a dokumentumok pedig a rekordok. A dokumentumok tartalmazhatnak alkollekciókat is. Adatbázisom két fő kollekciót tartalmaz: foods és users.

* 1. Foods kollekció

A foods (ételek) kollekció tartalmazza több, mint 5700 élelmiszer kalória és makrotápanyag értékeit. Ebből a kollekcióból kérdezünk le, amikor az alkalmazásban ételekre keresünk rá. A kollekció felépítése a 6.1. ábrán látható.

A képen asztal látható

Automatikusan generált leírás

6.1. ábra – Foods kollekció felépítése

Az egyes élelmiszerek dokumentumokként tárolódnak, és egyedi azonosítóval rendelkeznek. Minden élelmiszerről el van tárolva az élelmiszer megnevezése, kalóriáinak száma, hogy mennyi fehérjét, szénhidrátot és zsírt tartalmaz, az élelmiszer azonosítója, valamint a nameSearchField mező, ami tartalmazza az étel nevének szóközönként szétválasztott összes részszavát. A nameSearchField mező segítségével tudunk lekérdezni az adatbázisból. Ha rákeresünk egy étel nevére, akkor megnézi, hogy az szerepel-e valamelyik élelmiszernek a nameSearchField tömb elemei között. Ha igen, akkor van találat.

* 1. Users kollekció

A users kollekció tartalmazza a felhasználókról eltárolt adatokat: felhasználó személyes adatai, rögzített étkezések adatai, létrehozott receptek (6.2. ábra).

A képen szöveg látható

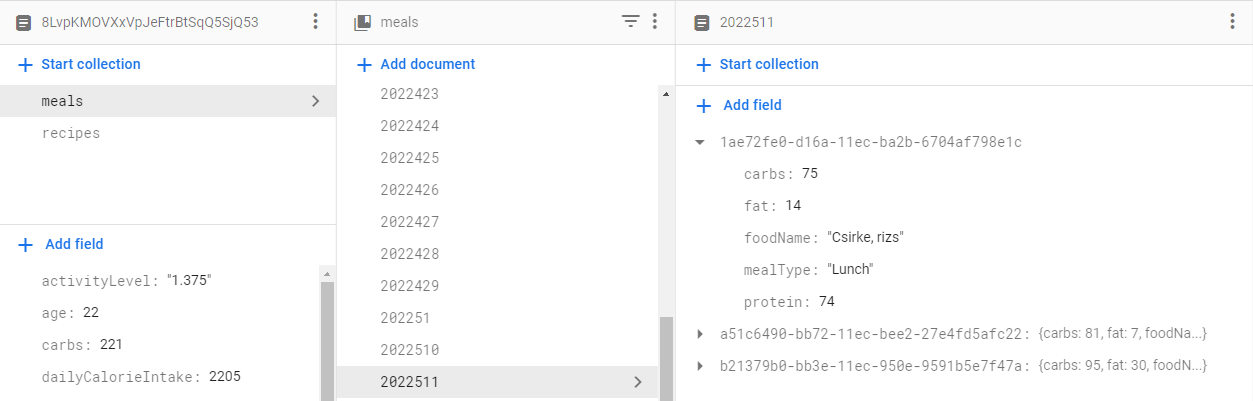
Automatikusan generált leírás

6.2. ábra – Users kollekció felépítése

A felhasználókról eltárolt személyes adatok a következők: név, nem, születési dátum, testsúly, magasság, aktivitási szint, cél, és a napi kalória- valamint makrotápanyag beviteli értékek. A users kollekció tartalmazza a meals és a recipes alkollekciókat.

* + 1. Meals kollekció

A meals kollekcióban vannak tárolva a felhasználó által rögzített étkezéseknek adatai.  
A 6.3. ábrán látható a kollekció felépítése.

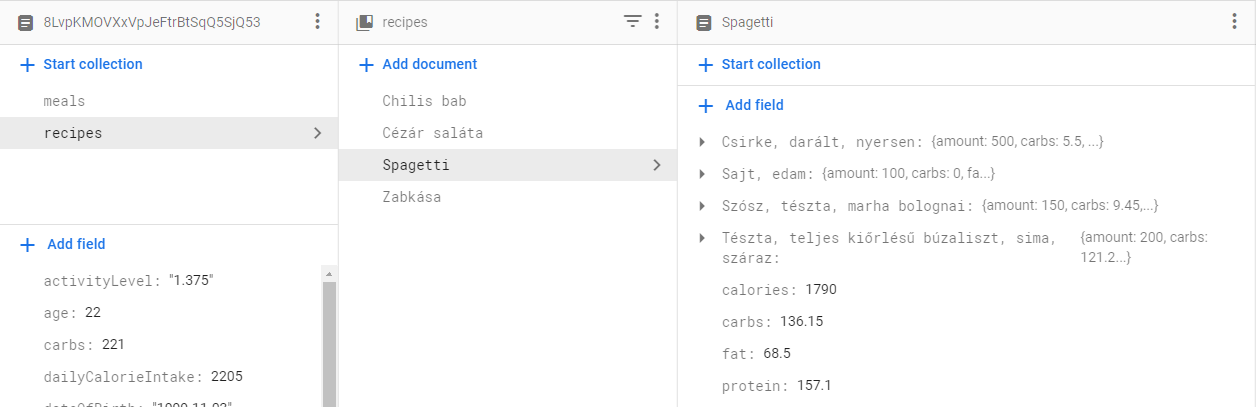


6.3. ábra – Meals kollekció felépítése

A kollekcióban szereplő dokumentumok dátumoknak felelnek meg. A dokumentum mezőit map-ek képezik, amik neve egy generált UUID egyedi azonosító. Minden mező egy-egy étkezésnek felel meg. Az étkezesről el van tárolva az elfogyasztott étel neve, az étkezés jellege, és a makrotápanyag értékei.

* + 1. Recipes kollekció

A recipes kollekcióban vannak tárolva a felhasználó által létrehozott receptek. A 6.4. ábrán látható a kollekció felépítése.



6.4. ábra – Recipes kollekció felépítése

A kollekcióban szereplő dokumentumok az egyes recepteket reprezentálják. A dokumentum azonosítója megegyezik a recept nevével. A dokumentumokban map-ek találhatók, amik a recept hozzávalóit reprezentálják, illetve a recept összesített kalória- és makrotápanyag értékei. A map-ek nevei megegyeznek a hozzávalók neveivel. Minden hozzávalóról el van tárolva a mennyisége, és a makrotápanyag értékei.

## Az étel adatbázis

Az alkalmazás egyik fő szempontja az volt, hogy rendelkezzen egy adatbázissal, amiben rá tudunk keresni szinte bármilyen létező élelmiszerre. Némi kutatás után rábukkantam a AUSNUT 2011–13 – Australian Food Nutrition Database-re. Ez az adatbázis az Ausztrália Új-Zéland Élelmiszer-szabványok által készült, és tartalmazza több mint 5700 élelmiszernek és italnak 51 fajta tápanyagértéket. Az adatbázis ingyenesen elérhető és szabadon felhasználható, tehát tökéletesen megfelel az én igényeimnek. Mivel az adatbázis Excel táblázat formájában áll rendelkezésünkre, ezért ezt át kellett alakítanom olyan formátumra, amit majd később importálni tudok a Firestore-ra.

A képen asztal látható

Automatikusan generált leírás

7.1. ábra - AUSNUT 2011–13 – Australian Food Nutrition Database

A 7.1. ábrán látható az Excel fájl megnyitva. Az ábrán nincs feltüntetve az összes oszlop, de így is látszik, hogy nagyon sok olyan információ szerepel az élelmiszerekről, amikre az alkalmazásom megvalósításához nem lesz szükségem. Ezért a következő lépés az volt, hogy kiválogattam a számomra szükséges mezőket, amik a következők voltak: étel azonosítója, étel neve, energia, fehérje, zsír, szénhidrát. Az étel energiája kJ (kilojoule) mértékegységben volt megadva, ezt át kellett alakítanom kcal-ra (kilokalória). Ez után egy weboldal segítségével lefordítottam a teljes adatbázist magyar nyelvre. Ahhoz, hogy az adatbázist fel tudjam tölteni a Firestore-ra, ahhoz JSON formátumban kell, hogy legyen. Ahhoz, hogy ezt meg tudjam tenni, előbb kiexportáltam az adatbázist egy CSV formátumú fájlba. Ez után egy weboldal segítségével a CSV formátumú fájlt át tudtam konvertálni JSON formátumba.

* 1. Keresőmező hozzáadása az adatbázishoz

A Firestore nem támogatja a full-text keresést, ami azt jelenti, hogy hiába van meg az adatbázisom, jelenlegi állapotában nem fogok tudni benne lekérdezni az ételek neve alapján. Ahhoz, hogy az adatbázisban történő keresést meg tudjam valósítani, létre kellett hoznom minden objektumra egy keresőmezőt. Ez a keresőmező tartalmazza az adott étel nevének szóközönként szétválasztott összes részszavát. Erre a mezőre ha ráállítunk egy array.contains filter-t, akkor a tömb elemei között meg fogja keresni az aktuális string előfordulását. Ahhoz, hogy az ételek neveit szétbontsam rész stringekre, ezeket a stringeket egy tömbbe rakjam, majd hozzáadjam az egyes étel objektumokhoz új mezőként, egy Node.js szkriptet használtam. A szkript kódja az [1] mellékletben megtalálható. Ezzel elkészült a tényleges JSON fájl, amit importálni kell a Firestore-ra.

* 1. JSON fájl feltöltése a Firestore-ra

A Firestore alapból nem biztosít semmilyen lehetőséget, hogy adatokat tudjunk importálni, ezért ezt is egy Node.js szkript használatával oldottam meg. Ahhoz, hogy szkript segítségével tudjunk adatokat feltölteni, Firebase Console-on előbb létre kell hozni a projektnak egy web appot. Majd ehhez a web apphoz létre kell, hogy hozzunk egy service account-ot, és végül generálnunk kell egy service key-t. Ez a service key ad engedélyt a szkriptnek, hogy írni tudjon a Firestore-ra. A szkript kódja a [2] mellékletben megtalálható.

## Irodalomjegyzék

[1] MyFitnessPal - <https://www.myfitnesspal.com/>

[2] YAZIO - <https://www.yazio.com/>

[3] LifeSum - <https://lifesum.com/>

[4] Macros - <https://play.google.com/store/apps/details?id=com.josmantek.macros&hl=en&gl=US>

[5] Does Calorie Counting Work? A Critical Look – Healthline - <https://www.healthline.com/nutrition/does-calorie-counting-work>

[6] Calorie – Wikipedia - <https://en.wikipedia.org/wiki/Calorie>

[7] Macro Calculator: Count Your Macros Like a Pro! – Bodybuilding - <https://www.bodybuilding.com/fun/macronutrients_calculator.htm>

[8] Calorie Calculator - <https://www.calculator.net/calorie-calculator.html>

[9] Macro Calculator - <https://www.calculator.net/macro-calculator.html>

[10] Mifflin St. Jeor Calculator – Leigh Peele - <https://www.leighpeele.com/mifflin-st-jeor-calculator>

[11] Flutter - <https://flutter.dev/>

[12] Pub.dev - <https://pub.dev>

[12] Firebase - <https://firebase.google.com/>

[13] Node.js - <https://nodejs.org/>

[14] Google Fonts - <https://fonts.google.com/>

[15] Food Standards Australia New Zealand (2014). AUSNUT 2011–13 – Australian Food Composition Database. Canberra: FSANZ. - <https://www.foodstandards.gov.au/science/monitoringnutrients/ausnut/foodnutrient/Pages/default.aspx>

[16] Free Online Translator - <https://www.onlinedoctranslator.com/en/translationform>

[17] Convert CSV to JSON - <https://www.convertcsv.com/csv-to-json.htm>

[18] Bulk Upload JSON Data to Cloud Firestore Collections -<https://www.youtube.com/watch?v=Qg2_VFFcAI8>

## Nyilatkozat

Alulírott Tóth Bagi Bence programtervező informatikus BSc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Szoftverfejlesztés Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Diplomamunka Repozitóriumában tárolja.

2022.05.14.

Aláírás

## Mellékletek

[1] Keresőmező kialakítását megvalósító szkript kódja

A képen szöveg látható

Automatikusan generált leírás

[2] JSON fájl Firestore-ra való feltöltését megvalósító szkript kódja

A képen szöveg látható

Automatikusan generált leírás