

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM



3kcal - kalóriaszámláló és követő alkalmazás

Tudományos Diákköri Konferencia Dolgozat

Témavezető:
dr. Novák Pálma-Rozália,
Egyetemi adjunktus

Végzős hallgatók:
Fülöp Bernadett-Timea,
Illés Izabella-Krisztina,
Nagy Henrietta

2025

Kivonat

A dolgozat témája a táplálkozás nyomon követésének hatékonyabbá tétele, különös tekintettel a kalóriabevitel gyors és pontos monitorozására. A táplálkozás napjainkban kiemelkedő szerepet játszik az egészséges életmód fenntartásában, melynek elősegítésére egyre többen próbálnak alkalmazni olyan lehetőségeket, amelyek elősegítik a kalóriabevitel pontosabb nyomon követését. A napi étkezések figyelemmel kísérése elengedhetetlen ahhoz, hogy személyre szabott táplálkozási terveket készíthessünk, különösen azok számára, akik a megfelelő testsúlyt vagy egyéb egészségi célokat szeretnék elérni. A kalóriák pontos követése azonban időigényes és gyakran bonyolult folyamat, különösen akkor, ha az étkezések összetett fogásokból állnak, és nem mindig könnyű meghatározni egy étkezés tápanyag-összetételét. Továbbá, az emberek hajlamosak alábecsülni az elfogyasztott kalóriákat, vagy időhiány miatt egyszerűen elfelejteni rögzíteni az étkezéseket, ami a pontos nyomon követés egyik legnagyobb akadályá.

A 3kcal egy innovatív alkalmazás, amely átfogó megoldást kínál az étkezések és a vízfogyasztás nyomon követésére. Az automatikus ételfelismerésnek köszönhetően a felhasználók gyorsan és egyszerűen rögzíthetik az elfogyasztott ételeket, valamint a Meal Line funkció segít az étkezések áttekinthető rendszerezésében. A beépített kalóriaszámláló rendszer valós idejű adatokat biztosít a napi bevitelről, megkönnyítve ezzel a pontos követést. A rendszer egy kezdeti Survey kitöltésével meghatározza a felhasználó napi kalóriaszükségletét és személyre szabott nutraciók ajánlásokat nyújt, ezzel támogatva az egészséges és kiegyensúlyozott táplálkozást.

Az alkalmazás Flutter technológiával készül, amely lehetővé teszi a többplatformos, rezponzív felhasználói felület kialakítását. A háttérrendszer egy mikroszervizes architektúrára épül, biztosítva a rugalmas skálázást és a hatékony működést. Az adatok tárolására és kezelésére több adatbázis-rendszert használunk, míg az alkalmazás biztonságát és konfigurációkezelését Vaultwarden biztosítja. A szolgáltatások kitelepítését és menedzselését Helm Chartok segítik, megkönnyítve a konténerizált környezetben történő üzemeltetést.

Kulcsszavak: kalóriabevitel, egészséges életmód, nyomon követés, táplálkozás, meal-line, ételfelismerés

Tartalomjegyzék

1. Bevezető	11
1.1. Célkitűzések	11
1.1.1. Fő célkitűzések	12
2. Elméleti megalapozás	13
2.1. Szakirodalmi áttekintő	13
2.1.1. Tápanyagok	13
2.1.2. Tápanyagkövetés	15
2.1.3. Tápanyagkövető alkalmazások	17
2.2. Piackutatás	18
2.2.1. Yazio	19
2.2.2. Calorie Mama	20
2.2.3. SnapCalorie	21
2.3. A napi kalóriabevitel és tápanyagok számítása	22
2.3.1. Alapanyagcsere és teljes napi energiafelhasználás (TDEE)	23
2.3.2. Cél alapú kalória módosítás	23
2.3.3. Makrotápanyagok számítása	24
2.3.4. Folyadékbevitel számítása	24
3. A rendszer specifikációja	25
3.1. Felhasználói követelmények	25
3.1.1. Use-case specifikációk	26
3.2. Rendszerkövetelmények	28
3.2.1. Funkcionális követelmények	28
3.2.2. Nem funkcionális követelmények	29
4. Tervezés és megvalósítás	34
4.1. Architektúra	34
4.2. Felhasznált technológiák	34
4.2.1. Backend technológiák	35
4.2.2. Mobile	36
4.2.3. DevOps	37
4.2.4. Mesterséges intelligencia modellek integrálása	38
4.3. Komponensek és funkciók	39
4.3.1. Felhasználói felület tervezése és megvalósítása	39
4.3.2. Felhasználói felület és funkciók	39

4.3.3.	Osztálydiagram	43
4.3.4.	Backend tervezése és megvalósítása	44
4.3.5.	Web-scraping	47
4.3.6.	Adatbázisok	47
4.4.	Kitelepítés	49
4.4.1.	CI-Pipeline	51
4.5.	Tesztelhetőség	54
4.6.	GitLab Issue Board	56
5.	A rendszer bemutatása	58
5.1.	Fő funkciók	58
5.1.1.	Bevitt tápanyagértékek és vízfogyasztás naplózása	58
5.1.2.	Kép alapú tápanyagérték felismerés	59
5.1.3.	Meal Line - Elfogyasztott ételek nyomon viuális követése	59
5.1.4.	Manuális tápanyagbevitel	60
5.1.5.	Receptek kezelése - CookBook	61
5.1.6.	Profil kezelés - saját adatok módosítása	62
6.	Összegzés	63
6.1.	Kinek ajánlott?	63
6.2.	Továbbfejlesztési lehetőségek	64
	Ábrák jegyzéke	65
	Táblázatok jegyzéke	66
	Irodalomjegyzék	68

1. fejezet

Bevezető

Felmerülhet bennünk az a kérdés, hogy mi is az a kalória, hogyan lehet és miért hasznos ezt számon tartani és odafigyelni rá. A kalória a táplálkozástudomány és energiafelhasználás terén az energia mértékegysége, amelyet az ételek és italok energiaértékének, valamint az emberi test energiafelhasználásának meghatározására alkalmaznak. Biológiai értelemben a kalória az egy energia, amit a szervezet az ételekben található szénhidrátokból, zsírokból és fehérjékből nyer, és amelyet a napi tevékenységekhez, például mozgáshoz, testhő fenntartásához, vagy akár a gondolkodáshoz használ fel.

Manapság egyre többen figyelnek arra, hogy a kalóriabevitel és az energiafelhasználás egyensúlyban legyen. Nem csoda, hiszen az elhízás és a vele járó egészségügyi problémák világszerte egyre nagyobb méreteket öltenek. Sokan ezért elkezdik nyomon követni a kalóriabevitelüket, mert úgy érzik, hogy ez hatékony módja lehet súlyuk kordában tartásának. A kalóriák nyomon követése azonban nem csak a fogyni vágyók számára fontos. A sportolók számára például elengedhetetlen, hogy pontosan tudják, mennyi energiára van szükségük a legjobb teljesítményükhöz. De hasznos lehet azoknak is, akik egyszerűen csak egészséges életmódot szeretnének élni, mivel segíthet nekik a megfelelő tápanyagarányok és energiaszintek fenntartásában a mindennapokban.

Jelen dolgozat és projekt lehetséges megoldást kínál arra, hogyan lehetne egyszerűsíteni és automatizálni a kalóriaszámlálást, hogy gyorsabb és kényelmesebb legyen. A cél az, hogy ne csak a napi kalóriabevitelt lehessen könnyen követni, hanem hogy a rendszer segítsen az étkezések megtervezésében, a tápanyagarányok helyes beállításában, és abban is, hogy elérjük a személyes céljainkat. Az automatizálás mellett az is fontos, hogy a folyamat érdekes és élvezetes legyen, így sokkal könnyebb lesz tudatosabban élni és odafigyelni az egészségünkre.

1.1. Célkitűzések

A projekt célja egy olyan kalóriatracker alkalmazás kifejlesztése, amely pontosan és hatékonyan képes nyomon követni a felhasználók étkezési szokásait, figyelembe véve a különböző diétás igényeket és életmódokat. Az alkalmazás célja a felhasználói élmény javítása, az étkezések gyors és egyszerű rögzítésével, valamint a személyre szabott tápanyagbevitel nyomon követésével.

1.1.1. Fő célkitűzések

- **Pontosság és széleskörű étkezési adatbázis:** Az alkalmazásnak átfogó és folyamatosan frissülő étkezési adatbázist kell biztosítania, amely a felhasználók számára pontos és releváns információkat nyújt az általuk fogyasztott ételek tápanyagtartalmáról. A folyamatos adatbázis-bővítés célja, hogy a felhasználók széles választékból választhassanak, és biztosak lehessenek abban, hogy az alkalmazásban szereplő információk naprakészek.
- **Testreszabhatóság és egyéni beállítások:** A felhasználóknak lehetőséget kell biztosítani arra, hogy egyéni étkezési szokásaikhoz és tápanyagszükségletükhöz igazítsák az alkalmazás működését. Ennek részeként testre szabhatják az adagokat, valamint hozzáadhatják azokat az ételeket, amelyek nem szerepelnek az alap adatbázisban, ezáltal biztosítva a pontos kalóriabevitelt és tápanyagok nyomon követését.
- **Különböző diéták támogatása:** Az alkalmazásnak képesnek kell lennie arra, hogy különböző étrendi preferenciákat támogasson, mint például a vegán, vegetáriánus, gluténmentes vagy alacsony szénhidráttartalmú diéták. Az alkalmazásnak személyre szabott étkezési terveket kell kínálnia, amelyeket a felhasználó életmódja és preferenciái alapján ajánl.
- **Motiváció és fejlődés követése:** Az alkalmazásnak segítenie kell a felhasználókat a céljaik elérésében, és folyamatos motivációt kell biztosítani számukra. Ehhez részletes grafikonok, statisztikák és fejlődési mutatók segítségével követhetik nyomon az étkezési szokásaikban és a tápanyagbevitelükben végbement változásokat. A rendszeres visszajelzések és célkitűzések segítenek a felhasználóknak abban, hogy elérjék a kívánt eredményeket.

Ezek a célkitűzések lehetővé teszik, hogy az alkalmazás egy komplex és felhasználóbarát eszközként szolgáljon a felhasználók számára, segítve őket az egészséges étkezési szokások kialakításában és fenntartásában.

2. fejezet

Elméleti megalapozás

A projekt célja egy olyan mobilalkalmazás kifejlesztése, amely a felhasználókat támogatja a táplálkozásuk és kalóriabevitelük folyamatos nyomon követésében. Mivel a megfelelő táplálkozás kulcsfontosságú szerepet játszik az egészséges életmódban, az alkalmazás célja, hogy segítse a felhasználókat a napi étkezéseik dokumentálásában, a tápanyagok és kalóriák pontos mérésében, valamint a táplálkozási céljaik elérésében. Az egészséges életmód fenntartása érdekében elengedhetetlen, hogy a felhasználók ne csak az étkezéseik minőségét, hanem azok kalóriatartalmát és tápanyagösszetételét is figyelemmel kísérik.

A mobilalkalmazás kifejlesztése során elengedhetetlen a tudományos és gyakorlati háttér megértése, hogy a felhasználók valóban hasznos, könnyen kezelhető és tudományosan megalapozott segítséget kapjanak. Ehhez szükséges a táplálkozástudományi alapok ismerete, valamint a kalóriaszámlálás jelentőségének és hatékonyságának megértése. Az ilyen típusú alkalmazások lehetőséget adnak arra, hogy a felhasználók tudatosan nyomon kövessék napi étkezéseiket, figyelemmel kísérik a napi kalória- és tápanyagbevitelüket, és így elérjék a kívánt eredményeket, legyen szó fogyásról, izomtömeg növelésről, vagy egyszerűen az egészséges táplálkozás fenntartásáról.

2.1. Szakirodalmi áttekintő

2.1.1. Tápanyagok

A táplálkozástudomány célja, hogy feltárja a tápanyagok és az emberi szervezet energiaigénye közötti összefüggéseket, miközben figyelembe veszi, hogyan támogatják ezek a test optimális működését és a különféle biológiai folyamatok fenntartását. Az energiaellátást három fő makrotápanyag biztosítja: szénhidrátok, fehérjék és zsírok, amelyek mindegyike egyedi szerepet tölt be az egészség megőrzésében.

Szénhidrátok

A szénhidrátok az elsődleges energiaforrásaink, minden grammból körülbelül 4 kcal energiát nyújtva [Org23c]. Ezek különösen fontosak olyan tevékenységekhez, amelyek gyors energiaellátást igényelnek, például intenzív mozgáshoz vagy szellemi koncentrációhoz. Az egyszerű szénhidrátok – például a cukrok – gyors energiát biztosítanak, de ezek túlzott fogyasztása vércukorszint-ingadozásokhoz és hosszú távon egészségügyi problémákhoz vezethet. Az összetett szénhidrátok, mint a teljes kiőrlésű gabonák, lassabban

bomlanak le, ezért hosszan tartó energiát biztosítanak és segítik a vércukorszint szabályozását [Org23b]. Ezen kívül a szénhidrátok hozzájárulnak a rostbevitelhez is, ami fontos a bélrendszer egészsége szempontjából.

Fehérjék

A fehérjék grammonként szintén 4 kcal energiát biztosítanak, de funkciójuk túlmutat az energiaszolgáltatáson: nélkülözhetetlenek az izomszövetek felépítéséhez és regenerációjához. Továbbá alapvető szerepük van az enzimek és hormonok előállításában, valamint az immunrendszer működésében [FO23]. A fehérjék minősége és forrása szintén fontos: állati eredetű és növényi fehérjék kombinációjával biztosítható az összes esszenciális aminosav bevétele. Sportolók, valamint idősebb felnőttek számára a megfelelő mennyiségű és minőségű fehérje fogyasztása elengedhetetlen az izomtömeg megőrzése érdekében.

Zsírok

A zsírok a legnagyobb energiatartalmú makrotápanyagok, grammonként 9 kcal energiát szolgáltatva [Org23c]. Bár a zsírok gyakran rossz hírnévnek örvendenek, elengedhetetlenek a hormontermeléshez, a sejtek működéséhez, és bizonyos vitaminok (A, D, E, K) felszívódásához. Az egészséges táplálkozás során azonban fontos figyelni a zsírbevitel minőségére: a telítetlen zsírsavak, például az olívaolajban és a halakban találhatóak, jótékony hatásúak, míg a telített és transzzsírok fogyasztását minimalizálni kell [Org23b].

Rostok és alkohol

A rostok nem szolgáltatnak energiát, de rendkívül fontosak az emésztőrendszer egészségének fenntartásában. Segítik a bélmozgást, csökkentik a koleszterinszintet és hozzájárulnak a szívbetegségek kockázatának csökkentéséhez [Org23c]. Az alkohol, bár grammonként 7 kcal energiát biztosít, nem tekinthető tápanyagnak, mivel hiányzik belőle bármilyen biológiai szempontból esszenciális tápanyag. Ráadásul túlzott fogyasztása káros hatással lehet a májra és gátolhatja más tápanyagok felszívódását [Org23a].

Energiaegyensúly

Az energiaegyensúly fenntartása létfontosságú az egészséges testsúly megőrzése és az életminőség javítása szempontjából. Ha az energiafelvétel meghaladja az energiafelhasználást, az súlygyarapodáshoz vezethet, míg az elégtelen energiafogyasztás alultápláltságot okozhat. Az egyéni szükségletekhez igazított, változatos étrend és rendszeres fizikai aktivitás kulcsszerepet játszanak a megfelelő energiaegyensúly fenntartásában [Org23c, Org23b].

Ez a megközelítés nemcsak az egészség megőrzését szolgálja, hanem hozzájárul a krónikus betegségek megelőzéséhez is. Az étkezési szokások tudatos alakítása tehát az egyik legfontosabb lépés a hosszú és egészséges élet felé. Különösen fontos ez olyan állapotok esetén, mint a cukorbetegség, a magas vérnyomás vagy a szív- és érrendszeri betegségek, ahol a táplálkozás és az energiaháztartás kiegyensúlyozása közvetlen hatással van a betegség lefolyására és az életminőségre.

2.1.2. Tápanyagkövetés

A tápanyagkövetés és a kalóriaszámlálás olyan folyamatok, amelyek célja az elfogyasztott ételek és italok mennyiségi és minőségi nyomon követése. Ezek az eszközök lehetővé teszik a napi tápanyagbevitel pontosabb ellenőrzését, ami hozzájárulhat az egészségesebb életmód kialakításához és fenntartásához. Az alapelvek a tápértékek, mint például a kalória, a makro- és mikrotápanyagok pontos dokumentálására és ezek fogyasztásának optimalizálására épülnek. A kalóriaszámlálás egy kvantitatív módszer, amely az elfogyasztott ételek energiatartalmának mérésére szolgál. Ennek alapja, hogy minden élelmiszer meghatározott mennyiségű energiát tartalmaz, amelyet kilokalóriában (kcal) fejeznek ki. Ez a módszer alapvető fontosságú az energia-egyensúly fenntartásában, amely az egészséges testsúly és az optimális anyagcsere működésének kulcsa.

A hagyományos, kézzel vezetett étkezési naplók mára sok esetben helyet adtak a digitális megoldásoknak. A mobilalkalmazások és online platformok jelentős előrelépést jelentenek a tápanyagkövetés terén, mivel automatikusan számolják az ételek kalória- és tápanyagtartalmát. A felhasználók könnyedén rögzíthetik étkezéseiket, míg a technológia képes elemezni az adatokat és személyre szabott tanácsokat nyújtani. A modern tápanyagkövető eszközök – például okostelefonok, okosórák és fitneszkarkötők – integrált szenzorokat és alkalmazásokat kínálnak az aktivitás, az energiafelhasználás és az étkezések monitorozására. Ezek a technológiai megoldások nemcsak kényelmesebbé teszik a nyomon követést, hanem pontosabb adatokat is szolgáltatnak a felhasználók számára. Az okosórák által mért aktivitási szintek például segíthetnek az energiafelhasználás pontosabb becslésében, míg az ételek beolvasására alkalmas alkalmazások automatikusan felismerik az ételek tápanyagtartalmát.

A tápanyagkövetés tudományos alapjai a humán anyagcsere és táplálkozástudomány eredményeire épülnek. A legfrissebb kutatások szerint a digitális tápanyagkövetés nemcsak az egyének egészségügyi céljainak elérését segítheti elő, hanem közegészségügyi szinten is fontos szerepet játszhat az elhízás és az alultápláltság kezelésében. A PubMed-en található tanulmányok szerint a digitális naplók alkalmazása jelentős mértékben növeli az önellenőrzés hatékonyságát, mivel az azonnali visszacsatolás motivációt adhat a felhasználóknak a jobb döntések meghozatalához.

A kalóriaszámlálás és tápanyagkövetés tudományos háttere, valamint pozitív és negatív hatásai széles körben kutatott terület, amely jelentős hatással van az egészségmegőrzésre és a táplálkozási szokások javítására. A WHO szerint a modern táplálkozási trendek és az energia-dús, feldolgozott élelmiszerek fogyasztásának növekedése hozzájárul az elhízás és más krónikus betegségek terjedéséhez [Org23d]. A kalóriaszámlálás olyan eszköz lehet, amely segít az embereknek megérteni az étkezési szokásaikat, és támogatja az egészséges életmód kialakítását azáltal, hogy elősegíti a kiegyensúlyozott étrend kialakítását és a túlfogyasztás elkerülését [Spr23].

Pozitív hatások

A tápanyagkövetés számos pozitív hatással jár, amelyek az egyéni és közösségi egészségügyi célok elérését egyaránt támogatják. Az egyik legfontosabb előnye, hogy segíti az embereket abban, hogy tudatosabbá váljanak az étrendjükre vonatkozóan. A táplálkozással kapcsolatos adatgyűjtés és az automatikus visszacsatolás révén a felhasználók jobban megérthetik, hogyan hatnak étkezési szokásaik az egészségükre. Ez a tudatos-

ság különösen fontos lehet az olyan népbetegségek, mint az elhízás vagy a cukorbetegség megelőzésében és kezelésében. A WHO jelentései rámutatnak arra, hogy a technológia, például mobilalkalmazások és viselhető eszközök, jelentős mértékben hozzájárulhatnak a súlykontroll fenntartásához és az egészséges életmód ösztönzéséhez.

Ezek az eszközök képesek személyre szabott ajánlásokat nyújtani, amelyek figyelembe veszik az egyéni preferenciákat, célokat és egészségügyi állapotot. Például egy alkalmazás nemcsak rögzíti a napi kalóriabevitelt, hanem segít az ételek optimális összetételének megtervezésében is, biztosítva ezzel a makro- és mikrotápanyagok kiegyensúlyozott bevitelét. Az ilyen típusú támogatás hatékonyan növeli a felhasználók motivációját és elkötelezettségét az egészséges életmód iránt.

Egy másik fontos pozitív hatás az életmódbeli változások nyomon követhetősége. A technológiai megoldások segítségével a felhasználók nemcsak az étkezési szokásaikat követhetik nyomon, hanem az aktivitásukat, alvásukat és egyéb egészségügyi mutatóikat is. Ezek az adatok integrálhatók, így komplex képet adhatnak az egyén egészségi állapotáról, amely elősegíti a hosszú távú fenntartható változásokat.

A közegészségügyi szinten a tápanyagkövető technológiák hozzájárulhatnak a lakosság egészségügyi állapotának javításához. Az adatok elemzése révén az egészségügyi szervezetek pontosabb képet kaphatnak az étkezési szokásokról és azok hatásairól, ami elősegítheti az egészséges táplálkozásra vonatkozó iránymutatások és politikák kidolgozását.

A tudományos kutatások folyamatosan megerősítik, hogy a kalóriaszámlálás és tápanyagkövetés akkor a leghatékonyabb, ha azt kombinálják egyéb egészségügyi stratégiákkal, mint például a rendszeres testmozgás, a stresszkezelés és az egészséges táplálkozásra való nevelés. Ezek az eszközök együttesen segíthetnek abban, hogy az emberek hosszú távon fenntartható életmódot alakítsanak ki, amely nemcsak az egészségükre, hanem a jólétükre is pozitív hatással van [Org23e].

Negatív hatások

Bár a tápanyagkövetés számos pozitív hatással bír, vannak bizonyos negatív aspektusai is, amelyeket figyelembe kell venni. Az egyik legjelentősebb kihívás a túlzott fókusz, amelyet az emberek az adatokra helyezhetnek. Ez különösen az olyan egyéneknél jelenthet problémát, akik hajlamosak a táplálkozással kapcsolatos kontroll túlhajtására. Az adataalapú megközelítés túlzott alkalmazása növelheti az ortorexia, vagyis az egészséges táplálkozással kapcsolatos kényszeres viselkedés kockázatát. Ez az állapot hosszú távon káros hatással lehet az érintettek mentális egészségére, mivel az étkezés öröme helyett az adatok és szabályok kerülnek a középpontba.

Egy másik jelentős probléma a felhasználói adatvédelem kérdése. A tápanyagkövető alkalmazások és eszközök rengeteg érzékeny személyes adatot gyűjtenek, például a táplálkozási szokásokról, egészségügyi állapotról és életmódról. Ha ezek az adatok nem megfelelően védettek, vagy rossz kezekbe kerülnek, az súlyos adatvédelmi incidensekhez vezethet. Továbbá, bizonyos felhasználók aggódhatnak amiatt, hogy ezek az adatok harmadik felekhez, például biztosítókhoz vagy marketingcégekhez kerülnek, ami etikai kérdéseket is felvet.

Technológiai szempontból a tápanyagkövető rendszerek pontossága és elérhetősége is problémát jelenthet. Sok eszköz és alkalmazás nem képes pontosan felismerni az ételek összetételét, ami téves adatokat eredményezhet, és félrevezető visszacsatolást adhat

a felhasználóknak. Továbbá, ezek a megoldások gyakran jelentős technológiai erőforrásokat igényelnek, például modern okostelefont vagy stabil internetkapcsolatot, amely nem mindenki számára elérhető. Ez társadalmi egyenlőtlenségeket eredményezhet, mivel az alacsonyabb jövedelmű csoportok kevésbé férhetnek hozzá ezekhez a szolgáltatásokhoz.

A kalóriaszámlálásnak negatív hatásai lehetnek, ha nem megfelelően használják. Például túlzott fókuszálás az energiabevitelre étkezési zavarok kialakulásához vezethet, mint például az orthorexia vagy az anorexia nervosa. Ezenkívül az alulértékelés, különösen a magas szénhidráttartalmú ételeknél, problémákat okozhat az étrend egyensúlyában. Az ilyen esetekben kulcsfontosságú az edukáció és a szakértői támogatás [Org23e, Org23d].

A WHO rámutatott, hogy az új technológiák – például a mobilalkalmazások – kulcs szerepet játszanak az egészséges életmód támogatásában. Ezek az eszközök lehetővé teszik az egyének számára, hogy pontosan nyomon kövessék napi kalóriabevitelüket és egyéb táplálkozási mutatóikat. Az elhízás elleni küzdelemben például a digitális egészségügyi alkalmazások fontos eszközként szolgálnak, mivel valós idejű adatokat szolgáltatnak és növelik a felhasználók tudatosságát.

2.1.3. Tápanyagkövető alkalmazások

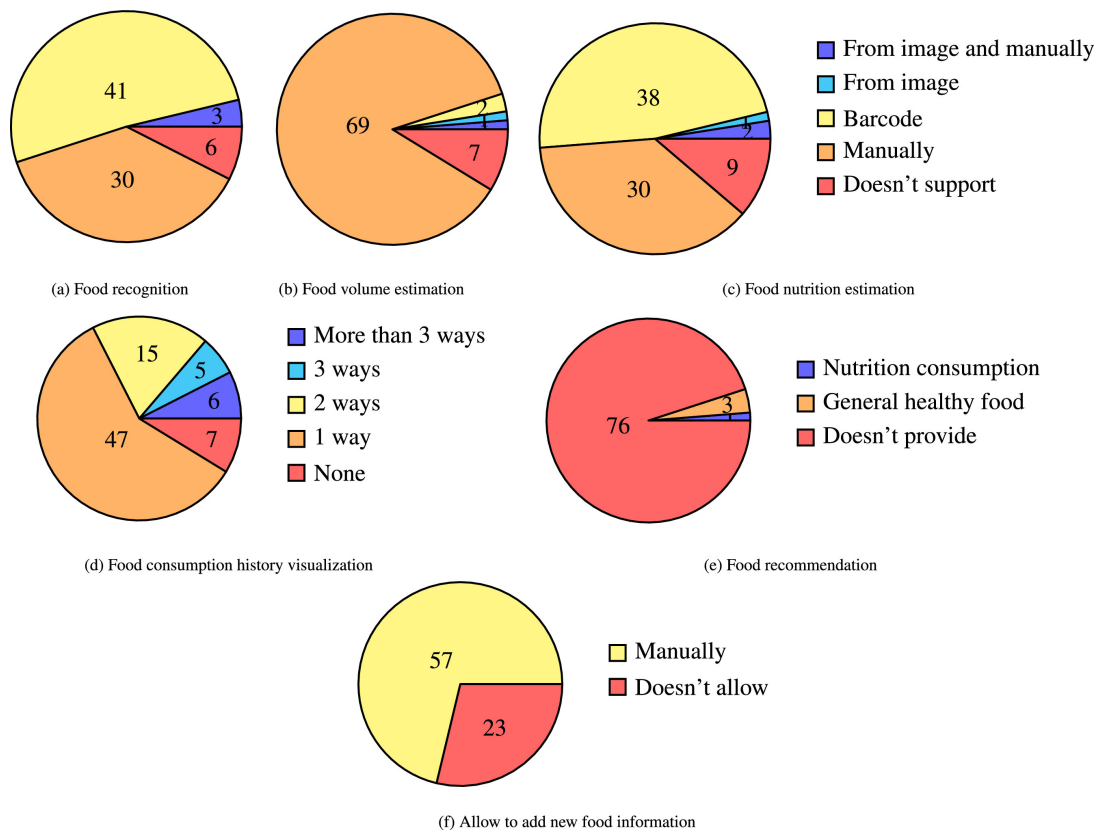
Az étel az emberi élet egyik alapvető szükséglete, amely nemcsak a túlélés eszköze, hanem az egészséges életmód alapja is. Az egészséges táplálkozás kulcsfontosságú a jól-lét szempontjából, azonban a rohanó életmód, rossz szokások és az önkontroll hiánya negatívan befolyásolják az ételválasztást. Az egészségtelen étrend, különösen a magas kalóriatartalmú és zsíros ételek fogyasztása, súlyos egészségügyi problémákhoz, például elhízáshoz vezethet, ami számos krónikus betegség, például magas vérnyomás, cukorbetegség és szív-érrendszeri betegségek kockázatát növeli.

Az elmúlt években a mobilalkalmazások, különösen az étkezési szokásokat nyomon követő és ajánlásokat adó appok elterjedése jelentősen megnőtt. Ezek az alkalmazások segítenek az egészséges életmód fenntartásában, például az ételek tápértékének kiszámításában vagy az étkezési szokások elemzésében. Az AI és a gépi tanulás technológiák, például a képfelismerés, lehetővé tették az ételek automatikus azonosítását és a táplálkozási értékek elemzését. Az ilyen technológiák nemcsak a fogyasztók étkezési döntéseit segíthetik, hanem az egészségügyi problémák, például az elhízás és a cukorbetegség kezelésében is fontos szerepet játszhatnak.

A kutatók különböző módszereket fejlesztettek ki az ételek azonosítására és a kalóriabevitel becslésére, beleértve a vizuális felismerést és a 3D rekonstrukciót. Emellett az ételajánlási rendszerek személyre szabott javaslatokat nyújthatnak, figyelembe véve a felhasználók egészségügyi állapotát, ízlését és étkezési szokásait. Ezek az alkalmazások jelentős hatással vannak a fogyasztók viselkedésére, és segítenek az egészségesebb ételválasztásban [Kes23].

Egy 2022-ben megjelent cikkben kritikus áttekintésre került 80 ételfogyasztást nyomon követő és ajánló alkalmazás, amely a három fő kereskedelmi alkalmazásáruházban (Google Play Store, Apple App Store és Microsoft Store) érhető el [SS22]. Az alkalmazásokat egy sajátos kifejlesztésű értékelőn vizsgálták meg, továbbá elemezve az alkalmazásokban található felhasználói véleményeket is, ezek nézőpontjainak és elvárásainak jobb megértése érdekében.

A kiértékelések szerint, és amint az a 2.1 ábrán is látható, a jelenlegi mobilalkalmazások többsége nem felel meg az ételfogyasztás nyomon követésére és ajánlásokra vonatkozó alapvető követelményeknek. Bár néhány alkalmazás tartalmazott bizonyos elvárt funkciókat, egyik sem biztosította az összes szükséges funkciót. A legtöbb esetben az adatok manuális bevitelére volt szükség, és az alkalmazások által használt adatbázisok nem elég gazdagok. Kevés bizonyítékokon alapuló alkalmazás létezik, annak ellenére, hogy számos kutatás foglalkozik az automatikus ételazonosítással, az adagméret becslésével és a tápérték számításával. Ezek olyan funkciók, amelyek egy modern alkalmazásokban mindenképp helyet kell kapniuk. Az ételajánlási funkció szintén hiányzik a legtöbb vizsgált alkalmazásból, ezért ez is fontos szempont a jövőbeli fejlesztések során. Mivel az alkalmazások diétás terveket javasolnak, ételeket ajánlanak és tápértékeket becsülnek, szakértő dietetikusok vagy táplálkozási szakértők bevonása elengedhetetlen. Emellett az adatbázisok gazdagítása is szükséges, hiszen manapság többféle ételmiszer-adathalmazt áll rendelkezésre. A szoftverminőségi szempontok, mint például az esztétika, általános funkciók, teljesítmény, használhatóság, szintén kulcsszerepet játszanak a kereskedelmi alkalmazásoknál, így a fejlesztőknek ezekre is figyelmet kell fordítaniuk [ZMKK⁺21].



2.1. ábra. Alkalmazások kiértékelése

2.2. Piackutatás

Ebben a fejezetben három népszerű kalóriakövető alkalmazást – a Yazio-t, a Calorie Mama-t és a SnapCalorie-t – mutatjuk be, kiemelve azok erősségeit és gyengeségeit.

Az alkalmazások elemzése során rámutatunk a leggyakoribb felhasználói problémákra, és javaslatokat teszünk arra, hogy ezek hogyan oldhatók meg egy fejlettebb, felhasználóbarátabb applikációval. A bemutatott problémák és javítási lehetőségek alapjául szolgálnak egy új alkalmazás fejlesztéséhez, amely célja a meglévő megoldások továbbfejlesztése és a felhasználói élmény optimalizálása.

2.2.1. Yazio

A kalóriatracking alkalmazások piaca az utóbbi években jelentős növekedésen ment keresztül, mivel a felhasználók egyre inkább a személyre szabott egészségügyi és táplálkozási adatok nyújtotta előnyöket keresik. Az egyik legismertebb ilyen alkalmazás a Yazio, amely számos funkcióval rendelkezik, de mint minden eszköz, neki is vannak gyengeségei. A következőkben bemutatjuk a Yazio alkalmazás főbb gyengeségeit, és azt, hogyan javíthatóak ezek az alkalmazásunkban.



2.2. ábra. Yazio főoldal

Adatbevitel pontossága

A Yazio automatikus adatfelismerési rendszere, amely az étkezések beírását segíti, nem mindig biztosítja a kívánt pontosságot. Egyes felhasználók szerint az alkalmazás nem mindig ismeri fel helyesen az étkezések tápanyagtartalmát, vagy nem találja meg az adott ételek pontos tápanyagainak adatait a rendszerben. Ez főként akkor fordul elő, ha az adatbázis nem tartalmazza az adott terméket vagy ha az adatok elavultak [App24].

Javítási lehetőség: A mi alkalmazásunkban fejlettebb képfelismerési algoritmusokat alkalmazhatunk, amelyek a felhasználók étkezéseiről készült képek alapján pontosabb tápanyag- és kalóriainformációkat szolgáltatnak. Továbbá folyamatosan frissíthetjük az adatbázist, hogy a lehető legnagyobb mértékben lefedje az étkezési lehetőségeket, beleértve a helyi ételeket és a nemzetközi termékeket is.

Felhasználói felület és élmény

Bár a Yazio egyszerű és áttekinthető felhasználói felülettel rendelkezik, egyes felhasználók panaszkodtak a navigáció bonyolultságára, különösen akkor, amikor új funkciókat kell használniuk. Az alkalmazás nem mindig nyújt intuitív élményt, különösen akkor, ha az új felhasználó nem ismeri jól az app funkcióit.

Javítási lehetőség: Az alkalmazásunkban hangsúlyt fektethetünk a felhasználói élményre (UX), olyan egyszerűsített navigációval és személyre szabott irányelvekkel, amelyek segítik a felhasználókat a funkciók gyors megtalálásában és használatában. A színvilág, a menürendszer és az interakciók is könnyebbé válhatnak.

Korlátozott étkezési adatbázis

Az egyik legnagyobb probléma, amire a felhasználók felhívták a figyelmet, az étkezési adatbázis korlátozott mérete és pontossága. A felhasználók gyakran jelzik, hogy a vonalkód beolvasásával nem találják meg a megfelelő ételleket, és gyakran előfordul, hogy az adatbázisban szereplő tétel nem felel meg a valóságnak. Ez különösen akkor zavaró, amikor más adatbázisokban elérhető tételek nem szerepelnek a Yaziban [Aim24].

Javítási lehetőség: A mi alkalmazásunkban folyamatosan frissíthetjük és bővíthetjük az étkezési adatbázist. Továbbá lehetőség lenne az étkezések adatainak valós idejű frissítésére, hogy a felhasználók mindig a legfrissebb információkat kapják.

Offline hozzáférhetőség

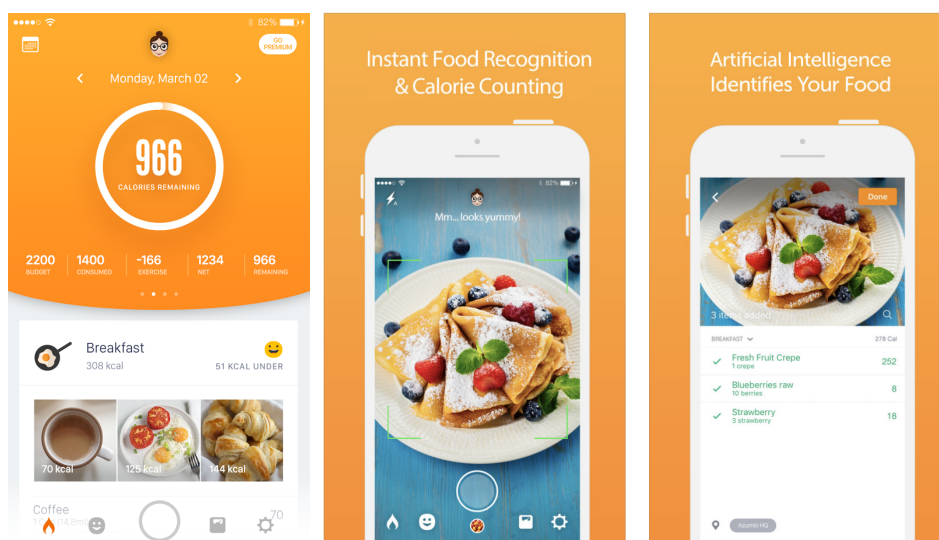
A Yazio alkalmazás az internetkapcsolatra épít, és nem biztosít teljes körű offline használatot. Ez különösen problémás lehet azoknak, akik olyan helyeken szeretnék használni az alkalmazást, ahol nincs elérhető internetkapcsolat.

Javítási lehetőség: Alkalmazásunkban egy offline módot fejleszthetünk, amely lehetővé teszi a felhasználók számára, hogy internetkapcsolat nélkül is rögzítsenek étkezéseket, majd később szinkronizálják az adatokat, amikor elérhetővé válik a kapcsolat.

2.2.2. Calorie Mama

A [Calorie Mama](#) egy innovatív kalóriaszámláló alkalmazás, amely mesterséges intelligenciát és képfelismerési technológiát használ az étkezések rögzítésére. Az alkalmazás különlegessége, hogy a felhasználóknak nem kell manuálisan megadniuk étkezéseiket, elegendő, ha egyszerűen fényképeznek egy ételt, és az alkalmazás azonnal képes azonosítani az ételt és kiszámítani annak tápértékét. Az AI-alapú rendszer nemcsak a kalóriákat, hanem a zsírokat, szénhidrátokat és fehérjéket is meghatározza, valamint a mennyiséget is megbecsüli a képek alapján [Kea17, cal24].

Az Calorie Mama legnagyobb előnye a könnyű használhatóság, mivel a felhasználóknak nem kell hosszú ideig keresgélniük az étkezések adatbázisában, mint sok más hagyományos étkezéskövető alkalmazásban. A képfelismerés és gépi tanulás algoritmusai lehetővé teszik, hogy az étkezéseket gyorsan és pontosan azonosítsák, még akkor is, ha azok nem szerepelnek az adatbázisban. Az alkalmazás AI szempontjából folyamatos fejlesztés alatt áll, így a felhasználók egyre pontosabb adatokat kapnak, miközben az étkezések nyilvántartása gyors és egyszerű [cal24, Car17].



2.3. ábra. Főoldal és képfelismerés

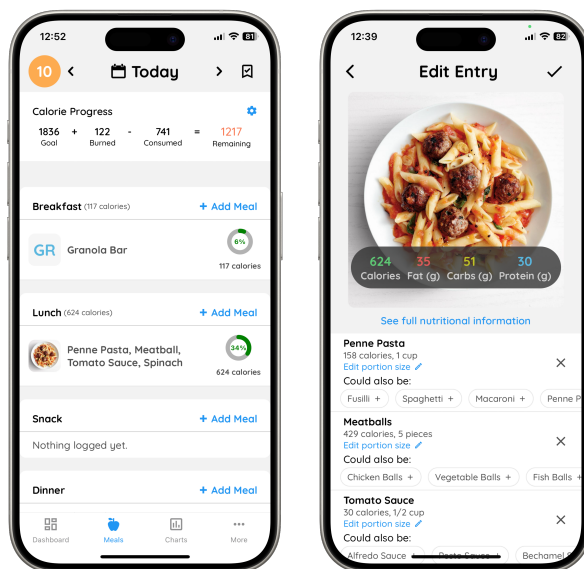
Javítási lehetőség: Az Calorie Mama előnye, hogy könnyen használható, de korlátozott a hosszú távú nyomon követés terén. Az alkalmazás nem biztosít részletes fejlődési mutatókat, így a felhasználók nem kapnak kellő visszajelzést a tápanyagbevitelükről és annak összefüggéséről a hosszú távú céljaikkal. Egy fejlettebb alkalmazás számára hasznos lenne, ha a felhasználók egyéni célokhoz és előrehaladásukhoz igazodó elemzéseket kapnának [cal24].

Továbbá a Calorie Mama nem biztosít olyan mélyreható napi visszajelzéseket, mint például a napi kalóriabevitel figyelemmel kísérése és a célokhoz való igazítás, ahogy azt a hagyományos étkezéskövető alkalmazások, mint a [MyFitnessPal](#) kínálják [Car17].

Javítási lehetőség: A prémium verzió további fejlesztései között szerepelhetne a felhasználói interakciók finomítása, például a tápanyagcéllok és napi kalóriabevitel vizuális ábrázolása, így még motiválóbba téve a felhasználói élményt.

2.2.3. SnapCalorie

A SnapCalorie egy mesterséges intelligenciával támogatott kalóriaszámláló alkalmazás, amelyet volt Google AI kutatók fejlesztettek. Az alkalmazás a Nutrition5k nevű adatbázison alapul, amely körülbelül 5000 ételfajtát tartalmaz. A felhasználók többféleképpen rögzíthetik étkezéseiket: képkészítéssel, hangalapú jegyzettel vagy vonalkód szkenneléssel. Emellett nyomon követhető benne a vízfogyasztás, makró- és mikrotápanyagok, valamint testreszabott ajánlásokat is nyújt.



2.4. ábra. SnapCalorie főoldal és adatbevitel oldal

Hozzáférés és funkcionalitás korlátozása

Az alkalmazás minden funkciója – beleértve az AI eszközöket – előfizetéshez kötött. A felhasználók szerint nincs ingyenes próbaidő, így nem tudják kipróbálni az app képességeit előzetesen.

Javítási lehetőség: Saját alkalmazásunk vendégként is biztosítja a főbb funkciókat, például a képalapú naplózást és a vízfogyasztás nyomon követését, így a felhasználók előzetesen is kipróbálhatják a szolgáltatásokat.

AI pontossága és eszközfüggőség

A SnapCalorie a LiDAR technológiát használja a kalóriabecsléshez, amely csak bizonyos iPhone modelleken érhető el. Más eszközökön a pontosság jelentősen csökkenhet, akár 265 kalóriás eltérést is okozva.

Javítási lehetőség: Az alkalmazásunk saját AI modellt használ, amely minden eszközön működik.

Felhasználói élmény és bejelentkezés

A SnapCalorie nem kínál harmadik fél általi (Google, Apple) bejelentkezést, a felület gyakran változik, és több automatikus funkció hibásan működik [Sto24].

Javítási lehetőség: A mi alkalmazásunk egyszerű, intuitív felületet kínál, valamint lehetőséget biztosít Google-fiókos bejelentkezésre, ezzel gyors és zökkenőmentes használatot biztosítva.

2.3. A napi kalóriabevitel és tápanyagok számítása

A napi kalóriabevitel és a makrotápanyagok szükséglete az egyéni jellemzők, mint például a testsúly, magasság, életkor és aktivitási szint figyelembevételével határozható

meg. A következő képletek és modellek segítenek az egyéni kalóriaszükséglet és tápanyag-eloszlás kiszámításában.

2.3.1. Alapanyagcsere és teljes napi energiafelhasználás (TDEE)

Az alapanyagcsere (BMR) kiszámítása a férfiak és nők számára különböző képletekkel történik:

- Férfiak esetén:

$$BMR_{\text{férfi}} = 66.5 + (13.8 \times \text{súly}) + (5 \times \text{magasság}) - (6.8 \times \text{életkor})$$

- Nők esetén:

$$BMR_{\text{nő}} = 655.1 + (9.6 \times \text{súly}) + (1.0 \times \text{magasság}) - (4.7 \times \text{életkor})$$

A teljes napi energiafelhasználás (TDEE) kiszámítása az alapanyagcserén (BMR) és az aktivitási szinten alapuló szorzaton történik. Az aktivitási szintet a következő faktorra becsüljük:

- Sedentary (ülő életmód): 1.0
- Low Active (alacsony aktivitás): 1.2
- Active (aktív életmód): 1.35
- Very Active (nagyon aktív életmód): 1.5

Az aktivitás szorzóval korrigált kalóriaigény:

$$TDEE = BMR \times \text{aktivitási faktor}$$

2.3.2. Cél alapú kalória módosítás

A napi kalóriabevitel további módosításra kerül az egyéni célok figyelembevételével. Az alábbi céloknak megfelelő szorzókat használunk:

- Súlycsökkentés: 0.85
- Súlymegőrzés: 1.0
- Súlynövelés: 1.1

A végső kalóriaérték a cél szorzó alkalmazásával alakul:

$$\text{Végső kalória} = BMR \times \text{aktivitási faktor} \times \text{cél szorzó}$$

2.3.3. Makrotápanyagok számítása

A napi kalóriabevitel alapján meghatározható a szükséges makrotápanyagok eloszlása. A tápanyagok arányait a célok (súlycsökkentés, súlymegőrzés, súlynövelés) alapján szabjuk meg:

- Fehérje (g):

$$\text{Fehérje} = \frac{\text{kalória} \times \text{fehérje arány}}{4}$$

- Zsír (g):

$$\text{Zsír} = \frac{\text{kalória} \times \text{zsír arány}}{9}$$

- Szénhidrát (g):

$$\text{Szénhidrát} = \frac{\text{kalória} \times \text{szénhidrát arány}}{4}$$

A fehérje, zsír és szénhidrát arányai a cél szerint változnak. Például, ha a cél a súlycsökkentés, a fehérje aránya 0.28, a szénhidrát aránya 0.43, míg a zsír aránya 0.29.

2.3.4. Folyadékbevitel számítása

A napi vízfogyasztás a testsúly függvényében történik, a következő képlettel:

$$\text{Folyadékbevitel (ml)} = 35 \times \text{testsúly (kg)}$$

Ezen felül, az aktivitási szint függvényében további folyadékbevitel szükséges. Az extra vízbevitel az alábbiak szerint módosul:

- Low Active: 300 ml
- Active: 600 ml
- Very Active: 1000 ml

Ezeknek a képleteknek az alkalmazásban való felhasználása biztosítja a személyre szabott, célorientált tápanyag- és folyadékbevitel pontos meghatározását.

3. fejezet

A rendszer specifikációja

3.1. Felhasználói követelmények

A felhasználói igények feltárása alapvető a sikeres alkalmazásfejlesztéshez. A felhasználók igényeinek megértése biztosítja, hogy az alkalmazás valóban megfeleljen a valós szükségleteiknek, javítva ezzel a felhasználói élményt és növelve az alkalmazás használatának gyakoriságát. A következő részben a felhasználói igényekre és azok kezelésére koncentrálunk, amelyek a felhasználó szemszögéből a legfontosabbak, valamint az ehhez kapcsolódó use-case diagrammok bemutatása következik.

A felhasználói élmény növelése érdekében figyelembe kell venni a következő fő szempontokat, amelyek közvetlenül befolyásolják az alkalmazás használhatóságát és hatékonyságát:

- **Egyszerű és intuitív felhasználói felület:** A felhasználók elvárják, hogy az alkalmazás könnyen használható legyen, és gyorsan elérjék a kívánt funkciókat. A felhasználói felület (UI) kialakításának egyszerűnek kell lennie, a navigáció pedig intuitív, hogy a felhasználók számára ne legyen zavaró az alkalmazás használata.
- **Személyre szabott étkezési ajánlások:** A felhasználók számára lehetőséget kell biztosítani arra, hogy személyre szabott étkezési tervet kapjanak, amely figyelembe veszi egyéni preferenciáikat, egészségi állapotukat és diétás igényeiket.
- **Részletes és pontos tápanyaginformációk:** A felhasználók elvárják, hogy az alkalmazás pontos és részletes információkat biztosítson az étkezések tápanyagtartalmáról.
- **Adatbevitel egyszerűsége:** Az alkalmazásnak lehetővé kell tennie az étkezések gyors és egyszerű rögzítését.

A fenti felhasználói igények összegzésére és vizualizálására hasznos egy **use-case diagram** alkalmazása, amely segít ábrázolni a felhasználói interakciókat és az alkalmazás főbb funkcióit. A use-case diagram segítségével világosabb képet kaphatunk arról, hogyan lépnek kapcsolatba a felhasználók az alkalmazással, és melyek azok a legfontosabb funkciók, amelyek közvetlenül befolyásolják az élményt és a hasznosságot.

A diagram a következő szereplőket és műveleteket tartalmazza:



3.1. ábra. Use-case diagram a felhasználói interakciókhoz

- **USER szerepkör:** A bejelentkezett felhasználó hozzáférhet a rendszer számos funkciójához, mint például a profilkezelés, vízfogyasztás nyomon követése, étkezések kiválasztása, valamint a napi log követése.
- **GUEST szerepkör:** A vendég felhasználó csak korlátozott funkciókat érhet el, mint a személyes adatok frissítése.

A use-case diagram az alkalmazás két fő felhasználói szerepkörét – USER (bejelentkezett felhasználó) és GUEST (vendég felhasználó) – mutatja be, valamint az ezekhez kapcsolódó funkciókat. A bejelentkezett felhasználók teljes hozzáféréssel rendelkeznek az alkalmazás funkcióihoz, beleértve a profilkezelést, az étkezések naplózását, vízfogyasztás követését, valamint a személyre szabott ajánlások és tápanyaginformációk megtekintését. Ezzel szemben a vendég felhasználók csak korlátozott funkcionalitást érnek el, például kipróbálhatják az alkalmazást vagy megadhatnak alapadatokat. A diagram vizuálisan segít megérteni, hogy a különböző felhasználók milyen módon lépnek kapcsolatba az alkalmazás különböző részeivel.

3.1.1. Use-case specifikációk

Bejelentkezés

Szereplők	Felhasználó
Use case neve	Google-fiókkal történő bejelentkezés
Leírás	A felhasználók kizárólag Google-fiókjuk segítségével tudnak bejelentkezni a rendszerbe.

Előfeltétel	A felhasználó rendelkezik érvényes Google-fiókkal, és az alkalmazást megnyitotta.
Utófeltétel	A felhasználó bejelentkezett a rendszerbe Google-fiókjával.
Fő folyamat	1. A rendszer megjelenít egy bejelentkezési oldalt, amelyen elérhető a „LOG IN” gomb. 2. A felhasználó rákattint a „LOG IN” gombra. 3. A rendszer átirányítja a Google bejelentkezési oldalára. 4. A felhasználó megadja Google-fiókjának adatait (ha még nincs bejelentkezve a Google-fiókjába). 5. A Google hitelesíti a felhasználót, és visszairányítja a rendszerbe. 6. A rendszer engedélyezi a hozzáférést a felhasználónak. A felhasználó teljes hozzáférést kap az alkalmazás minden funkciójához.
Alternatív útvonal	4. A felhasználó helytelen Google-fiókadatakat ad meg, vagy elutasítja a bejelentkezést. - A Google hibát jelez, és nem történik bejelentkezés. - A rendszer visszatér a bejelentkezési oldalra. Itt vagy újrapróbálja a bejelentkezést, vagy a "SKIP" gombra kattintva vendégként, tehát korlátozott hozzáférhetőséggel használja az alkalmazást.
Kivételek	- A Google-hitelesítés sikertelen a szerver vagy hálózati hiba miatt. - A felhasználó nem rendelkezik Google-fiókkal. - A felhasználó úgy dönt, hogy csak körülnézni szeretne az alkalmazáson, és vendég üzemmódban szeretne belépni. Ezt megteheti a „SKIP” gombra kattintva.

3.1. táblázat. Google-fiókkal történő bejelentkezés use case specifikációja

3.2. Rendszerkövetelmények

3.2.1. Funkcionális követelmények

A funkcionális követelmények olyan konkrét irányelvek, amelyek leírják a szoftvernek vagy rendszernek végrehajtandó viselkedést, funkciókat és műveleteket. A szoftver- és rendszerfejlesztés során ezek a követelmények körvonalazzák, mit kell tennie a rendszernek a felhasználói igények és az üzleti célok kielégítése érdekében.

Felhasználói regisztráció és bejelentkezés

Az alkalmazás kizárólag Google-fiókkal történő bejelentkezést tesz lehetővé, ezzel biztosítva a gyors és biztonságos hitelesítést. A felhasználónak nem szükséges külön fiókot létrehoznia, így nincs szükség külön jelszó megadására vagy annak visszaállítására. A Google-fiókos bejelentkezés leegyszerűsíti a regisztrációs folyamatot, és lehetővé teszi az adatok automatikus szinkronizálását a felhőalapú tárolással.

Felhasználói profil kezelése

A felhasználó képes módosítani a kitöltött kérdőív alapján megadott adatokat, beleértve a célt (például súlycsökkentés, izomnövelés stb.), életkort, magasságot, súlyt, nemet és életmódot. A módosítások után a rendszer automatikusan újraszámolja a napi kalóriabevitelt és a folyadékbevitel szükségletét.

Étkezések és tápanyagtartalom rögzítése

A felhasználó képes étkezéseket rögzíteni az alkalmazásban, beleértve az étkezés nevét, a hozzávalókat, és azok tápanyagtartalmát. A rendszer lehetővé teszi az étkezések gyors hozzáadását fotóalapján vagy manuálisan történő bevitellel. A felhasználók könnyen hozzáadhatják az étkezéseket, és az alkalmazás automatikusan kiszámítja azok tápanyagtartalmát.

Napi log és nyomon követés

A rendszer naplózza a felhasználó étkezéseit és tápanyaginformációit egy napi logban. A felhasználó könnyen áttekintheti a napi étkezéseit, és nyomon követheti a tápanyagbevitelét. Az alkalmazás biztosítja a napi logok vizualizálását, amely segíti a felhasználókat a céljaik elérésében és a diétás előírások betartásában.

Vízfogyasztás nyomon követése

A rendszer lehetőséget biztosít a felhasználó napi vízfogyasztásának nyomon követésére. A felhasználó beállíthat napi vízfogyasztási célokat, amit át is léphet, ha sikerül nagyobb mennyiségű vizet bevinni a szervezetébe.

Célok és előrehaladás nyomon követése

A felhasználó képes célokat kitűzni, mint például napi kalóriabevitel, makrotápanyagok (fehérje, zsír, szénhidrát) bevitele, valamint vízfogyasztás, és az alkalmazás folyamatosan nyomon követi és elemzi a célok elérését.

3.2.2. Nem funkcionális követelmények

Teljesítménykövetelmények

A 3kcal alkalmazás hatékony működéséhez kiemelten fontos a gyors válaszidő, a magas áteresztőképesség és az optimális erőforrás-használat.

A Flutter keretrendszer gyors válaszidőt biztosít, különösen a fejlesztési fázisban a hot-reload funkciónak köszönhetően. Az alkalmazás teljesítménye különböző platformokon és eszközökön is optimalizált, figyelembe véve a CPU és GPU terhelést. Az alacsonyabb teljesítményű eszközökön történő tesztelés garantálja, hogy a felhasználói élmény következetesen jó legyen.

A rendszer nagy mennyiségű tranzakció kezelésére alkalmas, amit a Kubernetes-alapú skálázás és a Traefik terheléelosztó biztosít. Egyidejű adatfeltöltések és -letöltések során az adatbázis replikáció és a MySQL optimalizálás minimalizálja a válaszidőt, míg a háttérszolgáltatások közötti aszinkron kommunikáció csökkenti a hálózati terhelést.

Az alkalmazás optimalizált CPU-, memória- és akkumulátorhasználatot kínál. A háttérben futó funkciók, például az értesítések kezelése, alacsony erőforrás-igénnyel működnek. A Docker-konténerek és a Kubernetes erőforrás-limitjei biztosítják, hogy egyetlen komponens se használja túl a rendelkezésre álló kapacitást.

Megbízhatóság

A megbízhatóság kulcsfontosságú az olyan alkalmazások számára, mint a ThreeKcal, amelyek felhasználói adatokkal dolgoznak, és napi szinten segítik a felhasználók étkezéseinek nyomon követését. A megbízhatóságot az alábbi három fő területen kell biztosítani: hibatűrés, rendelkezésre állás, valamint adatmentés és helyreállítás. Ezek az elemek biztosítják az alkalmazás zavartalan működését és a felhasználók adatainak biztonságát, még váratlan körülmények között is.

Az alkalmazás kezelni tudja az offline állapotokat, hálózati hibákat és hibás adatbevitelt anélkül, hogy összeomlana. Offline módban az adatok lokálisan tárolódnak, majd kapcsolat helyreállítása után szinkronizálódnak a felhővel. Hálózati hibák esetén a Kubernetes és Traefik újrapróbálkozási mechanizmusokat alkalmaz, a bemenetek validációja pedig megelőzi a hibás adatok okozta problémákat.

A folyamatos elérhetőséget a Kubernetes automatikus helyreállítása és a Traefik terheléelosztó biztosítja. Az alkalmazás több replikával fut, így hiba esetén más példány veszi át a működést. A health check funkciók folyamatosan figyelik a szolgáltatások állapotát.

Az adatbázisok automatikus mentése és a Persistent Volume Claims megoldások biztosítják az adatok biztonságos tárolását és visszaállítását. Új eszköz esetén a felhasználói adatok automatikusan szinkronizálódnak a felhőből, a mentések pedig titkosítva vannak tárolva.

Biztonság és hozzáférés-vezérlés

Az alkalmazás biztonságos hitelesítési és hozzáférés-vezérlési megoldásait az OAuth 2.0 szabvány segítségével valósítjuk meg. Az OAuth 2.0 széles körben elterjedt iparági szabvány, amely lehetővé teszi a felhasználók számára, hogy harmadik fél szolgáltatásait, például a Google-t használják a biztonságos hitelesítéshez. Az alkalmazásban a passport-google-oauth20 stratégia segítségével integráltuk a Google OAuth szolgáltatását, amely egy gyors és biztonságos bejelentkezést biztosít.

A hitelesítési folyamat során a felhasználó a bejelentkezési képernyőn keresztül a Google opciót választja, majd a Google beleegyezési képernyőjén engedélyezi az alkalmazás számára az alapvető profiladatokhoz (pl. email, név) való hozzáférést. Sikeres bejelentkezés után a Google visszairányítja a felhasználót az alkalmazás visszahívási URL-jére, ahol az alkalmazás a profiladatok alapján létrehozza vagy frissíti a felhasználó adatait az adatbázisban. Ezek az adatok magukban foglalják a felhasználó Google-azonosítóját, nevét, email címét és egyéb opcionális mezőket, mint például célkitűzéseit, életkorát vagy testsúlyát.

Az alkalmazás különböző jogosultsági szintek kezelésére is alkalmas, ugyanis megkülönböztet bejelentkezett felhasználót és vendéget is. A passport modulban található isAuthenticated funkció lehetővé teszi, hogy csak hitelesített felhasználók férhessenek hozzá a védett erőforrásokhoz és funkciókhoz. Ezt például a /protected végponton is alkalmazzuk, ahol csak a sikeresen bejelentkezett felhasználók kapnak hozzáférést.

Az alkalmazás további biztonsági rétegeket biztosít a session-kezelés révén. Az express-session middleware használatával a munkamenetek kezelése titkosított sütiken keresztül történik. A sütik biztonságát tovább növeli a secure opció, amely biztosítja, hogy a munkamenet sütik csak HTTPS-kapcsolaton keresztül működjenek.

A mobilalkalmazás szintén integrált OAuth 2.0 hitelesítést használ a felhasználók bejelentkeztetésére. A Flutter alkalmazás speciális végpontokon, például a flutterLogin és flutterGetUser útvonalakon keresztül kommunikál a Node.js alapú backenddel. A flutterLogin végpont fogadja a felhasználók által generált Google ID token-t, amelyet az alkalmazás ellenőriz és azonosít. Sikeres azonosítás esetén a felhasználó adatai frissítésre kerülnek az adatbázisban, majd a backend munkamenetet hoz létre a felhasználó számára, amely biztonságos hozzáférést biztosít a védett erőforrásokhoz.

A Google Developer Console-ban az alkalmazás számára létrehoztunk egy OAuth beleegyezési képernyőt, amely részletesen tájékoztatja a felhasználókat az adatkezelés részleteiről, így biztosítva az átláthatóságot. Az itt konfigurált OAuth kliens azonosító és titok garantálja, hogy csak az engedélyezett alkalmazás férjen hozzá a felhasználói adatokhoz.

Skálázhatóság

A modern alkalmazások egyik alapvető követelménye a skálázhatóság, amely biztosítja, hogy a rendszer képes legyen a növekvő terheléssel és felhasználói igényekkel lépést tartani. Alkalmazásunk skálázhatósága több szinten valósul meg, beleértve a horizontális és vertikális skálázást, konténerizált megoldások használatát Dockerben és Kubernetes környezetben, valamint egy jól definiált mikroszolgáltatás-architektúrát.

A horizontális skálázhatóság az alkalmazás backend komponenseinek több szerverre történő elosztását jelenti. Ez a megközelítés lehetővé teszi, hogy a rendszer dinamikusan

bővüljön a terhelés növekedésével, elkerülve az egyetlen szerver túlterhelését. A Docker és Kubernetes infrastruktúrák alapját képezik az alkalmazás horizontális skálázhatóságának.

A rendszerben minden szolgáltatás külön konténerben fut, amelyeket egy Docker Compose fájl definiál. A különböző komponensek – például a nutrition-management, user-management, és image-processor – mindegyike saját konténerben van elhelyezve, és külön adatbázissal dolgozik. A konténerek közötti kommunikációt a threecal_network hálózat biztosítja, míg az adatbázisok egészségét healthcheck szkriptek figyelik, hogy biztosítsák a szolgáltatások zavartalan működését.

A Kubernetes-alapú üzemeltetés során a rendszer további rugalmasságot nyer. A komponensek deployment fájlok segítségével kerülnek telepítésre, amelyek lehetővé teszik a replikációt és automatikus skálázást. A Kubernetes Traefik Ingress Controller segítségével biztosítja a bejövő kérések megfelelő elosztását, miközben az SSL-tanúsítványokat a Cert-Manager kezeli, garantálva a biztonságos kommunikációt.

A vertikális skálázhatóság a meglévő szerverek erőforrásainak növelését jelenti, például nagyobb memória vagy gyorsabb processzor hozzáadásával. Az alkalmazás MySQL alapú adatbázisai esetében, mint a threecal-db és user-management-db, fontos, hogy ezek a szerverek elegendő erőforrással rendelkezzenek a nagy mennyiségű adat tárolására és kezelésére.

A Kubernetes lehetővé teszi az erőforrások hatékony allokációját és dinamikus bővítést. Az adatbázisok és szolgáltatások konténerizált futtatásának köszönhetően könnyen frissíthetők a magasabb teljesítményigényű rendszerekre. A Kubernetes konfigurációkban megadható resource limits és requests, amelyek segítenek a különböző komponensek optimális működésének fenntartásában.

Karbantarthatóság és modularitás

A Flutter alkalmazás egyik alapvető erőssége a widget-alapú architektúrája, amely lehetővé teszi a projekt könnyű moduláris felépítését. Az alkalmazás különböző funkciói és képernyői különálló widgetekre építhetők, amelyek függetlenül fejleszthetők és tesztelhetők. Mivel a Flutter minden UI elemet widgetként kezel, ez segít abban, hogy az alkalmazás különböző részei egymástól függetlenül működjenek. A widgetek könnyen újrahasznosíthatók, módosíthatók és bővíthetők anélkül, hogy az alkalmazás más részein kódváltoztatásokat kellene eszközölni.

A pubspec.yaml fájlban deklarált függőségek segítenek abban, hogy a projekt folyamatosan bővíthető legyen anélkül, hogy az alapszerkezetet meg kellene változtatni. Az alkalmazás különböző részei — például a felhasználói felület, az adatkezelés, vagy az integrációs tesztek — jól körülhatárolt modulokban dolgozhatnak. Az egyes funkciók és komponensek közötti kommunikációt jól definiált API-k (például provider alapú állapotkezelés) irányítják, amelyek biztosítják az átláthatóságot és a könnyű karbantartást.

A main.dart fájl felépítése szorosan összefügg a projekt modularitásával. A különböző képernyők és funkciók, mint például a LandingPage, Survey, és NonUserPage, egyértelműen meghatározott route-ok segítségével vannak elkülönítve, ami lehetővé teszi a kód átláthatóságát és bővíthetőségét. A MultiProvider használata révén a különböző adatkezelő modellek, például a felhasználói adatok (UserProvider), a kérdőívek (SurveyProvider), és a tápanyagmérleg (NutrientsModel), jól elkülönítve és könnyen karbantarthatóan működnek. A ChangeNotifierProvider-ek segítenek biztosítani, hogy az alkalmazás

különböző részei hatékonyan kommunikáljanak egymással, miközben az egyes modellek önállóan tesztelhetők és fejleszthetők.

A `main.dart` fájl struktúrája tehát nemcsak a projekt modularitását segíti elő, hanem a karbantarthatóságot is. Az alkalmazás különböző részeinek, mint például a felhasználói adatok vagy a tápanyagmérleg kezelése, külön modellekben történik, így a jövőbeli fejlesztések gyorsan beilleszthetők az alkalmazásba anélkül, hogy az egész kódot át kellene alakítani. A route-ok átlátható kezelése, ahol minden képernyő és funkció világosan definiált, lehetővé teszi az új képernyők és funkciók egyszerű integrálását.

Ezen kívül a `flutter_lints` csomag használata segít abban, hogy a kód minősége és a stíluskonvenciók egységesek legyenek, így minden fejlesztő számára világosak a szabályok. A `analysis_options.yaml` fájlban beállított linter szabályok további biztosítékot adnak arra, hogy a fejlesztők moduláris és jól strukturált kódot írjanak, amely egyszerűen karbantartható és bővíthető.

Továbbá a backend részen a `pyproject.toml` és az `eslint.config.mjs` fájlban szereplő beállítások, mint a `pylint` és `black` konfigurációk, valamint a `@eslint/js` csomag használata nemcsak a kód formázására és minőségére vonatkozóan adnak irányelveket, hanem arra is ösztönzik a fejlesztőket, hogy egyértelmű és olvasható kódot írjanak. Az alkalmazás fejlesztéséhez használt eszközök és konfigurációk, mint például a provider alapú állapotkezelés és a jól meghatározott route-ok, mind hozzájárulnak a projekt könnyű karbantartásához és bővítéséhez.

Felhasználói élmény (UX)

Az alkalmazás felületének intuitívnak kell lennie, és minden felhasználó számára könnyen tanulható módon kell működnie. A Flutter widget-alapú struktúrája lehetővé teszi, hogy az alkalmazás különböző képernyői és funkciói vizuálisan is jól elkülönüljenek, miközben a navigáció egyszerű és érthető marad. A `main.dart` fájlban szereplő route-ok, mint például a `LandingPage`, `Survey`, és `NonUserPage` világosan meghatározzák a navigációs utakat, amelyek segítenek abban, hogy a felhasználó könnyedén eligazodjon az alkalmazás különböző részei között. A különböző képernyők közötti váltás zökkenőmentes, így a felhasználók gyorsan megtanulják, hogyan használják az alkalmazást, anélkül hogy bonyolult műveletekhez kellene folyamodniuk.

A Flutter alkalmazás dizájnya rendkívül testreszabható, lehetővé téve az esztétikai elemek egyszerű módosítását anélkül, hogy az alkalmazás funkcionalitását befolyásolnánk. A `flutter_native_splash` csomag például gondoskodik arról, hogy a betöltési képernyő vizuálisan is vonzó legyen, ezáltal az alkalmazás indulása is simább és élvezetesebb élményt nyújt. A színek, ikonok és egyéb vizuális elemek használata az alkalmazásban összhangban áll a tervezett UX irányelvekkel, amelyek célja, hogy a felhasználók gyorsan megértsék a funkciók működését.

Az alkalmazás sebessége és válaszkészsége szintén fontos tényező a felhasználói élmény javításában. A Flutter keretrendszer biztosítja, hogy az alkalmazás gyorsan reagáljon a felhasználói interakciókra, minimális késleltetéssel. Az alkalmazás minden egyes widgetje optimalizálva van, hogy a lehető leggyorsabban töltődjön be és reagáljon, miközben biztosítja a sima görgetést és a gyors válaszütemet. A `MultiProvider` használata, amely az alkalmazás adatállapotát kezeli, biztosítja, hogy az állapotváltozások hatékonyan frissüljenek.

Interoperabilitás

Az interoperabilitás kulcsfontosságú szempont, amikor az alkalmazás más rendszerekkel és szolgáltatásokkal való együttműködését kell biztosítani. Az alkalmazás fejlesztése során érdemes figyelembe venni, hogy a felhasználók igényei folyamatosan változnak, és előfordulhat, hogy harmadik fél szolgáltatásait kell integrálni, hogy az alkalmazás további funkciókkal bővüljön. Ezért az alkalmazásnak képesnek kell lennie más rendszerek és szolgáltatások, például analitikai eszközök, közösségi média API-k vagy más külső adatforrások egyszerű és hatékony integrálására.

A `flutter_dotenv` csomag segítségével az alkalmazás konfigurálható úgy, hogy különböző külső API-k elérhetőségeit és hitelesítő adatait környezeti változókból olvassa be, így biztosítva az API-k dinamikus kezelését. A `.env` fájl használata a projekt biztonságosabbá tételét is szolgálja, mivel nem szükséges érzékeny adatokat közvetlenül a kódban tárolni.

A harmadik fél szolgáltatásainak integrálása nem csupán az adatkommunikáció megoldásában segít, hanem az alkalmazás felhasználói élményének gazdagításában is. Például egy közösségi média API, mint a `google_sign_in` csomag, lehetővé teszi, hogy a felhasználók egyszerűen bejelentkezzenek a Google-fiókjukkal, míg a `http` csomag segíthet a hálózati kéréskezelésben, így az alkalmazás könnyedén kapcsolatba léphet külső szerverekkel, vagy adatokat kérhet le és küldhet.

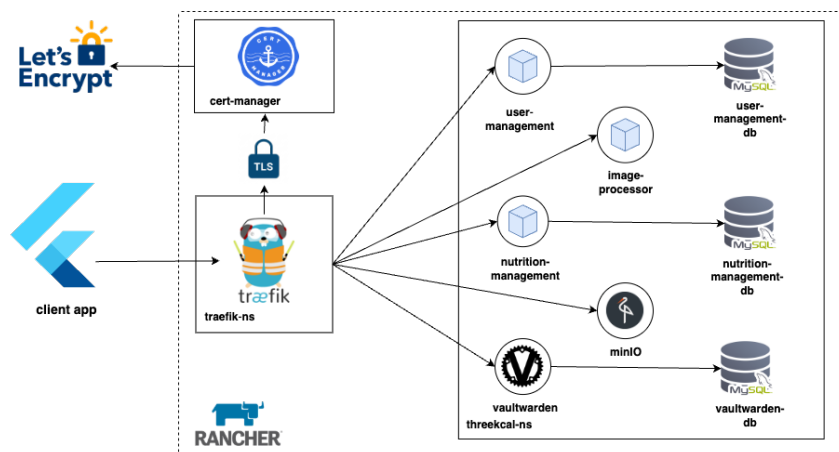
A más rendszerekkel való kompatibilitás szoros összefüggésben áll az alkalmazás skálázhatóságával és jövőbeli bővíthetőségével. Mivel a Flutter támogatja a platformok közötti fejlesztést (Android, iOS, Web), így az integrált API-k is minden platformon megfelelően működhetnek, biztosítva a sima felhasználói élményt.

4. fejezet

Tervezés és megvalósítás

4.1. Architektúra

Projektünkben mikroszolgáltatás-alapú architektúrát alkalmaztunk annak érdekében, hogy a rendszer könnyen skálázható, rugalmasan bővíthető és jól karbantartható legyen. Ez a megközelítés lehetővé teszi az egyes komponensek elkülönült fejlesztését és üzemeltetését, miközben biztosítja azok hatékony együttműködését egy integrált rendszerként..



4.1. ábra. Architektúra diagram

A 4.1 ábra az alkalmazás moduláris felépítését ábrázolja, ahol az egyes komponensek külön konténerekben futnak, és Kubernetes-alapú környezetben vannak kezelve.

4.2. Felhasznált technológiák

A technológiai stack olyan technológiák, keretrendszerek és eszközök összessége, amelyeket az alkalmazás fejlesztése és üzemeltetése során használunk. Az adott projektben különböző komponensekhez más-más technológiát alkalmaztunk, a backend fejlesztéséhez pedig **Node.js** és **Python** alapú megoldásokat vettünk igénybe. A frontend fejlesztéséhez a **Flutter** keretrendszert választottuk. Az adatok tárolására relációs adatbázisokat, konkrétan **MySQL**-t használtunk.

Az üzemeltetés során olyan modern technológiák nyújtottak támogatást, mint: **Docker**, **Kubernetes**, **Cert-Manager**, **Let's Encrypt**, **Traefik**.

Ezek kombinációja biztosította a rugalmasságot, skálázhatóságot és a projekt magas szintű teljesítményét.

4.2.1. Backend technológiák

Nutrition Management (Node.js)

A táplálkozással kapcsolatos adatok kezelését a **Node.js** környezetben valósítottuk meg. Ez egy aszinkron működésre optimalizált JavaScript futtatási környezet, amely különösen hatékony REST API-k fejlesztésére.

Felhasznált könyvtárak:

- **Express**: Gyors és egyszerű webalkalmazás-fejlesztési keretrendszer REST API-k létrehozásához.
- **Sequelize**: ORM (Object-Relational Mapping) eszköz a MySQL adatbázisok kezeléséhez.
- **dotenv**: Konfigurációs adatok (pl. API kulcsok, adatbázis-hozzáférési adatok) biztonságos tárolására.
- **jsonwebtoken**: Biztonságos token-alapú hitelesítés megvalósítására.
- **cors**: Az API-k közötti biztonságos kommunikáció támogatására.
- **mysql2**: Gyors és hatékony MySQL adatbázis-kezelő eszköz.

User Management (Node.js)

A felhasználókezelés funkcióját szintén **Node.js** alapokon fejlesztettük ki, amely biztosítja a felhasználói hitelesítést, valamint az adatok kezelését.

Felhasznált könyvtárak:

- **passport** és **passport-google-oauth2**: A Google OAuth integrációhoz használt hitelesítési megoldás.
- **express-session**: Felhasználói munkamenetek kezelésére.
- **bcrypt**: Jelszavak biztonságos tárolása és ellenőrzése.

Image Processing (Python)

A képfeldolgozási feladatokat – például az élelmiszerek azonosítását és elemzését – **Python** nyelven valósítottuk meg a **Flask** keretrendszer segítségével.

Felhasznált könyvtárak:

- **Flask**: Könnyen használható, skálázható webalkalmazás-fejlesztési keretrendszer.
- **requests**: HTTP-kérelmek küldésére és fogadására szolgáló könyvtár.
- **inference-sdk**: Képfeldolgozási modellek alkalmazásához szükséges könyvtár.
- **python-dotenv**: Környezeti változók biztonságos kezelésére.

4.2.2. Mobile

A projekt mobilalkalmazása Flutter keretrendszerben készült, amely lehetővé teszi a natív élményhez hasonló többplatformos fejlesztést. Az alkalmazás fejlesztése során különböző csomagokat (package-eket) használtunk, amelyek segítségével az alkalmazás funkcionalitását hatékonyan és strukturáltan alakítottuk ki, figyelembe véve a teljesítményt, biztonságot és felhasználói élményt.

Felhasználói felület és design:

- **cupertino_icons**: iOS-stílusú ikonok biztosítása az alkalmazás számára.
- **carousel_slider**: Képek és egyéb tartalmak elforgatható megjelenítéséhez használt csúszka komponens.
- **percent_indicator**: Haladásjelzők és százalékos kijelzések megjelenítésére szolgál.

Hitelesítés és biztonság:

- **google_sign_in**: Google-fiókkal való bejelentkezés és hitelesítés megvalósítása.
- **dotenv, flutter_dotenv**: Környezeti változók kezelése, API-kulcsok és egyéb érzékeny adatok tárolására.

Adatkezelés és hálózati kommunikáció:

- **http**: HTTP kérések küldésére és fogadására az API-kkal való kommunikáció érdekében.
- **path, path_provider**: Helyi fájlrendszer elérése és fájlok tárolásának támogatása.
- **shared_preferences**: Egyszerű lokális adattárolás biztosítása, például felhasználói beállítások mentésére.
- **intl**: Nemzetközi dátum- és időformázás támogatása.

Állapotkezelés:

- **provider**: Az alkalmazás állapotának hatékony kezelésére és adatmegosztásra különböző widgetek között.

Multimédia és eszközhozzáférés:

- **camera**: Kamerakezelés a képrögzítési funkciókhoz.
- **image_picker**: Képek kiválasztása a galériából vagy kamera használatával történő rögzítés.

Adatmodell és JSON kezelés:

- **json_annotation, json_serializable, build_runner**: JSON adatmodell létrehozása és sorosítás támogatása.

Tesztelés és fejlesztési eszközök:

- **flutter_test, integration_test**: Automatizált tesztelés biztosítása.
- **mockito, network_image_mock, fake_http_client**: Mock adatok létrehozása és API hívások tesztelése.
- **test, test_cov_console**: Egyéni tesztelési folyamatok lefuttatása és tesztlefedettség elemzése.

4.2.3. DevOps

Docker

A **Docker** egy konténerizációs platform, amely lehetővé teszi az alkalmazások és azok függőségeinek egyetlen csomagba történő összezsugorolását. A konténerek biztosítják az alkalmazás platformfüggetlenségét és könnyű átvitelét fejlesztői, tesztelési és éles környezetek között. A projekt során a backend szolgáltatásokat Docker image-ek formájában csomagoltuk, és a következő eszközöket használtuk:

Felhasznált eszközök:

- **Docker Compose** – Több konténeres alkalmazások együttes futtatása és kezelése.
- **Docker Hub** – Publikus és privát Docker image-ek használata.

Kubernetes

A **Kubernetes** a konténerizált alkalmazások telepítésére és kezelésére szolgáló platform. A Kubernetes lehetővé teszi az alkalmazások automatizált skálázását és magas rendelkezésre állásának biztosítását. A rendszerünk három backend microservice-t és két adatbázist futtat egy Kubernetes klaszteren belül.

Felhasznált eszközök:

- **Helm** – Csomagkezelő Kubernetes alkalmazások telepítéséhez és kezeléséhez.
- **Artifact Hub** – Helm chartok telepítése és frissítése.

Cert-Manager + Let's Encrypt

A **Cert-Manager** és a **Let's Encrypt** segítségével biztosítjuk az automatikusan megújuló SSL/TLS tanúsítványokat Kubernetes környezetben. Ez lehetővé teszi a titkosított kommunikációt minden microservice számára.

Felhasznált eszközök:

- **Cert-Manager** – Kubernetes beépülő modul a tanúsítványok automatikus kezeléséhez.
- **Let's Encrypt** – Ingyenes SSL/TLS tanúsítványok generálása és megújítása.

Traefik

A **Traefik** egy dinamikusan konfigurálható terheléselosztó, amely kifejezetten Kubernetes és Docker alapú rendszerekhez készült. Automatikusan felismeri a Kubernetes szolgáltatásokat és biztosítja a HTTPS terminálást Cert-Manager segítségével.

4.2.4. Mesterséges intelligencia modellek integrálása

A rendszer két különböző funkcióban is alkalmaz mesterséges intelligenciát: az ételfelismerés automatizálására, valamint az alapanyagok adminisztratív ellenőrzésére és ki egészítésére.

Ételfelismerés

Az alkalmazás az ételfelismeréshez a **LogMeal** által kínált ingyenes API-t használja, amely lehetővé teszi, hogy a felhasználók által feltöltött ételfotók alapján automatikusan azonosítsuk az ételt. A LogMeal modellje képes többféle ételtípus felismerésére képek alapján, és visszaadja azok nevét, kategóriáját, valamint becsült tápanyagtartalmát.

Ez különösen hasznos, amikor a felhasználó nem manuálisan szeretné rögzíteni a napi étkezését, hanem egyszerűen egy fotóval szeretné azt dokumentálni. Az ételfelismerés így nemcsak kényelmesebbé, hanem gyorsabbá is teszi az adatbevitel folyamatát.

Superadmin funkciók helyettesítése Ollamával

A felhasználók számára biztosított a lehetőség új alapanyag (ingredient) hozzáadására az adatbázishoz. Ennek során az **Ollama** nevű lokálisan futó mesterséges intelligencia-modell végzi el az alapanyag automatikus validálását. Ez azt jelenti, hogy amikor a felhasználó új hozzávalót kíván rögzíteni, az Ollama megvizsgálja az alapanyag megadott adatait (pl. név, leírás, kategória), és eldönti, hogy azok megfelelőek-e az adatbázisba történő beszúráshoz.

Amennyiben az ellenőrzés sikeres, az alapanyag bekerül az adatbázisba, így későbbiekben már a rendszer részeként használható lesz. Az Ollama emellett automatikusan generálja az adott alapanyag 100 grammra vetített tápértékeit (kalória, fehérje, zsír, szénhidrát), ezzel segítve a pontos tápanyagkövetést és a felhasználói élmény növelését.

Ez a megközelítés jelentősen csökkenti az adminisztratív feladatokat, és biztosítja, hogy az adatbázisba csak validált és pontos adatok kerüljenek be, a rendszer működésének integritása érdekében.

4.3. Komponensek és funkciók

4.3.1. Felhasználói felület tervezése és megvalósítása

Követelmények meghatározása

Az alkalmazás fejlesztése az alábbi alapvető követelmények alapján indult:

- **Célcsoport meghatározása:** Egészségtudatos felhasználók, akik szeretnék pontosabban követni táplálkozásukat.
- **Alapfunkciók:**
 - Kalóriabevitel és tápanyagkövetés.
 - Ételfelismerés képalapú technológiával.
 - Receptek böngészése és étkezésenkénti kategorizálása.
 - Személyre szabott napi kalóriaajánlás a felhasználói adatok alapján.
- **Felhasználói élmény (UX):** Letisztult, minimalista UI design, egyszerű, intuitív navigáció, gyors betöltési idő és reszponzív működés.

Tervezési eszközök

A felhasználói felületek és az alkalmazás áramlásának megtervezéséhez **Figma** szoftvert használtunk. A következőket készítettük el:

- **Interaktív prototípusok:** Az alkalmazás képernyőinek és funkcióinak vizuális bemutatása.
- **Iteratív fejlesztés:** A prototípusokat folyamatosan frissítettük a tesztelés során gyűjtött visszajelzések alapján.

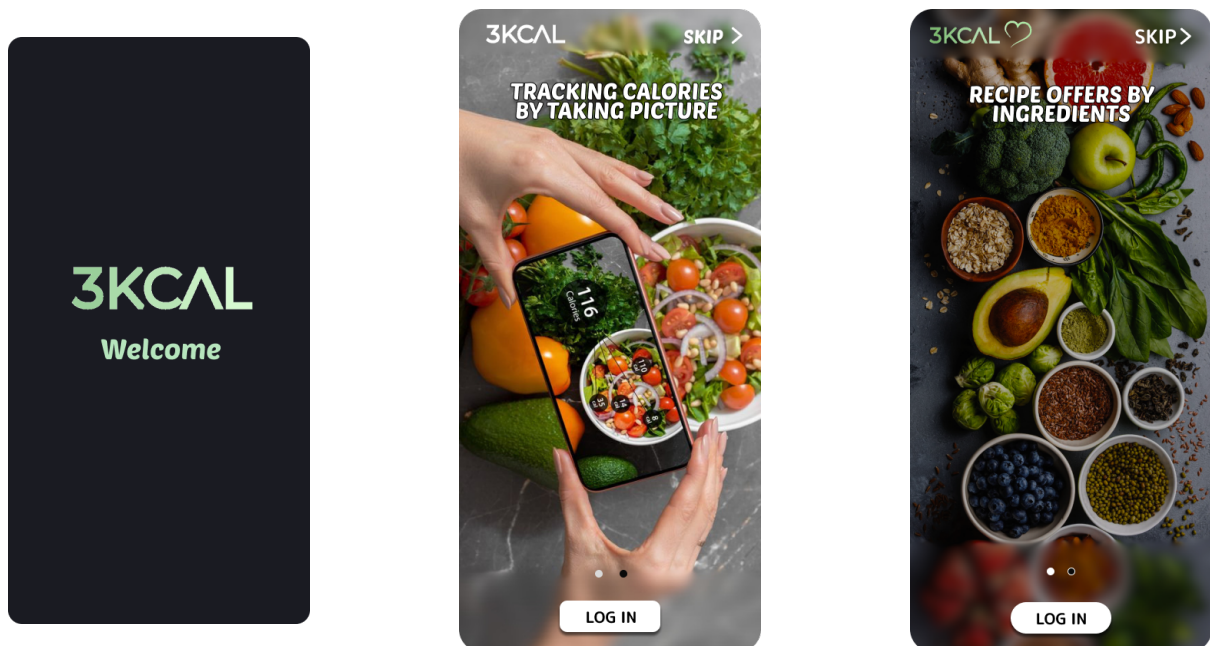
4.3.2. Felhasználói felület és funkciók

Splash képernyő

Az alkalmazás elsődleges indítóképernyője egy **egyszerű, minimalista Splash Screen**, amely bemutatja az alkalmazás logóját és egy üdvözlő üzenetet. A képernyő célja, hogy már az induláskor pozitív első benyomást keltsen a felhasználókban.

Bevezető képernyők

A bevezető szakasz célja, hogy bemutassa az alkalmazás alapvető funkcióit: a kép alapú ételfelismerést és annak kalóriaszámontartását, illetve receptajánlást.

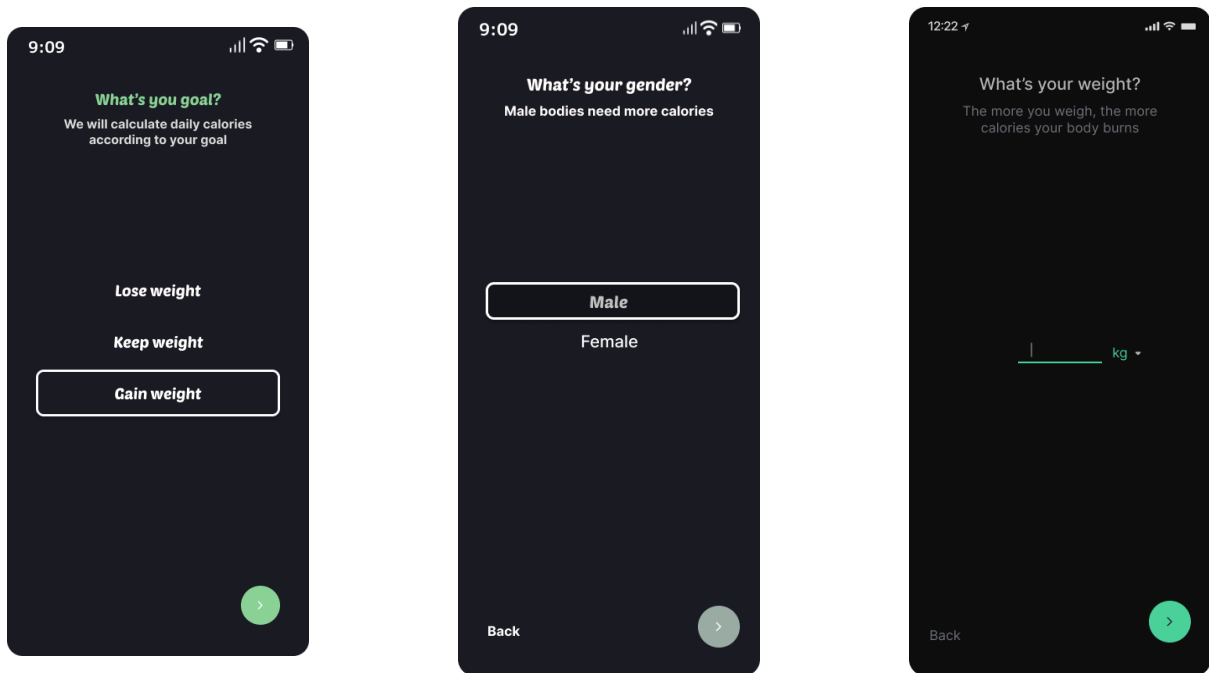


4.2. ábra. Splash képernyő (bal oldal), első bevezető oldal (középen), második bevezető oldal (jobb oldal).

Felhasználói adatok megadása

Az alkalmazás testreszabása érdekében a felhasználóknak meg kell adniuk néhány alapvető adatot:

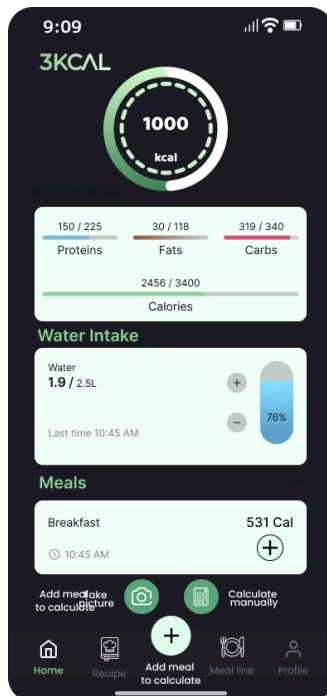
- Cél: például fogyás, súlytartás, hízás.
- Nem meghatározása.
- Aktivitási szint: például ülő életmód, aktív életmód.
- Magasság és testsúly beírása.



4.3. ábra. Cél meghatározása (kép bal oldalon), nem kiválasztása (középen), testsúly megadása (jobb oldal).

Főképernyő (Main Screen)

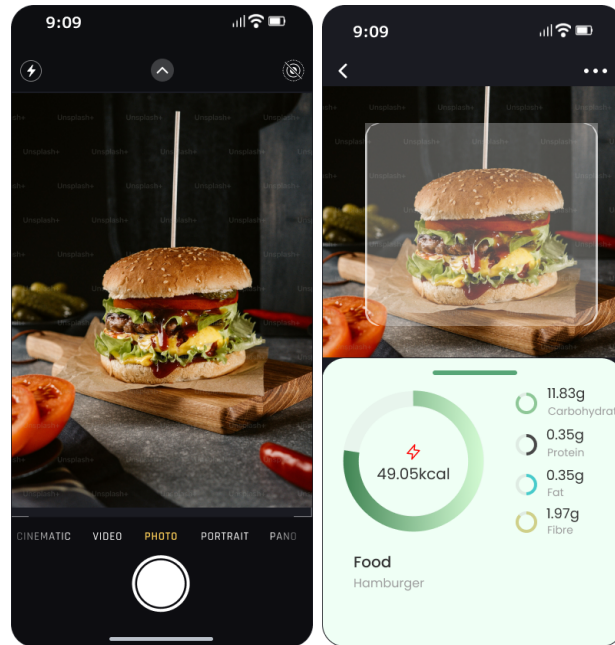
A főképernyő a napi kalóriefogyasztás és a haladás áttekintésére szolgál.



4.4. ábra. A főképernyő UI terve

Képfelismerés és ételfelismerés

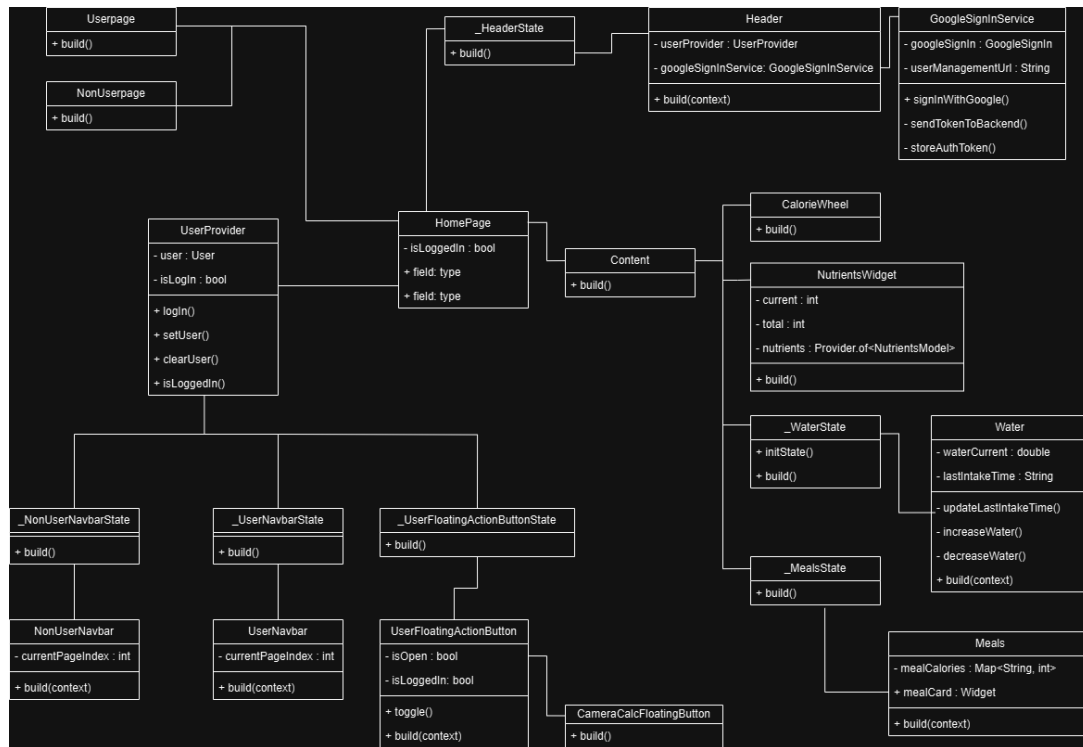
A képfelismerő rendszer az alkalmazás egyik leginnovatívabb funkciója. A kamera segítségével készült képek elemzésével az ételek azonosíthatók és kalóriaértékeik megjeleníthetők.



4.5. ábra. Kamera megjelenítése(bal oldal), képfelismerés, illetve a felismert étel adatai(jobb oldal)

4.3.3. Osztálydiagram

Home Page



4.6. ábra. Home Page osztálydiagram

A megadott osztálydiagram a Home Page rendszerének osztályait és azok közötti kapcsolatokat mutatja be. A főoldal célja, hogy a felhasználók számára különböző funkciókat és információkat biztosítson, mint például navigáció, kalória- és vízbevitel nyomon követése, valamint további tartalmak elérése. A diagram moduláris felépítése lehetővé teszi az egyes funkciók elkülönített fejlesztését és integrálását. Az alábbiakban részletesen bemutatom a komponenseket és azok működését.

A főoldal központi eleme a HomePage osztály, amely a felhasználói állapotot (isLoggedIn: bool) kezeli, és további mezőket tartalmaz, amelyek a főoldal tartalmát definiálják. Ez az osztály a fő komponensek kiindulópontja, és biztosítja a különböző funkciók integrálását. Ehhez szorosan kapcsolódik a navigációs rendszer, amelyet az UserNavbar és a NonUserNavbar osztályok valósítanak meg. Az UserNavbar a bejelentkezett felhasználók számára biztosít navigációs funkciókat, míg a NonUserNavbar a be nem jelentkezett felhasználók számára kínál lehetőségeket. Mindkét osztály a build(context) metódust használja a megfelelő felhasználói felület megjelenítésére. Az ezekhez tartozó állapotkezelő osztályok, mint például a _UserNavbarState és a _NonUserNavbarState, az aktuális oldal indexének kezelésére szolgálnak (currentPageIndex: int).

A fejléc, amely a főoldal tetején helyezkedik el, a Header osztály által valósul meg. Ez az osztály tartalmazza a felhasználói információkat, valamint a Google-bejelentkezés funkcióját. A Google-hitelesítés kezelésére szolgáló GoogleSignInService osztály lehetőséget nyújt a Google-fiókokkal való hitelesítésre. Az ehhez kapcsolódó funkciók közé tarto-

zik a `signInWithGoogle()`, amely elindítja a hitelesítési folyamatot, a `sendTokenToBackend()` metódus, amely a hitelesítési tokent továbbítja a háttér szolgáltatásnak, valamint a `storeAuthToken()`, amely helyben tárolja a tokent. A fejléc állapotának kezeléséért a `HeaderState` felel, amely biztosítja a dinamikus megjelenítést a `build()` metóduson keresztül.

A főoldal tartalmaiért a `Content` osztály felel, amely a különböző modulokat fogja össze, például a kalória- és vízbevitel, valamint az étkezések kezelésére szolgáló elemeket. A `CalorieWheel` modul a kalóriafogyasztás vizuális megjelenítését valósítja meg, például grafikon formájában, amely a `build()` metódussal kerül kirajzolásra. Ehhez kapcsolódik a `NutrientsWidget`, amely a makrotápanyagok (például fehérje, zsír, szénhidrát) adatait jeleníti meg. Ez az osztály tárolja az aktuális tápanyagokat (`current`), az összes elfogyasztott tápanyagot (`total`), valamint egy modellt (`nutrients`), amely a tápanyagadatokat kezeli. A megjelenítés itt is a `build()` metódus segítségével történik.

A vízbevitel nyomon követését a `Water` modul végzi. Az aktuális vízfogyasztást a `waterCurrent` mező tárolja, míg az utolsó vízfogyasztási időt a `lastIntakeTime` mező rögzíti. Az ehhez kapcsolódó funkciók közé tartozik az `updateLastIntakeTime()` a fogyasztási idő frissítésére, az `increaseWater()` és `decreaseWater()` a vízfogyasztás növelésére vagy csökkentésére. A megjelenítés szintén a `build(context)` metódussal történik. Az étkezésekkel kapcsolatos funkciókat a `Meals` osztály valósítja meg, amely a napi étkezések kalóriabevitelét kezeli. A `mealCalories` mező egy térképként tárolja az egyes étkezések kalóriaértékeit, míg a `mealCard` mező a grafikai megjelenítésért felel. A `build(context)` metódus itt is a modul vizuális elemeinek kirajzolását végzi.

A felhasználói interakciók kezelésében kulcsfontosságú a `UserFloatingActionButton`, amely egy lebegő gombot valósít meg. Ez az osztály kezeli a gomb nyitott állapotát (`isOpen: bool`), valamint a felhasználó bejelentkezési állapotát (`isLoggedIn: bool`). A `toggle()` metódus váltogatja a gomb állapotát (megnyitás vagy zárás). A `CameraCalcFloatingButton` osztály ezen lebegő gomb egy speciális megvalósítása, amely például egy kamera-alapú kalóriaszámító funkciót kínálhat. A megjelenítés szintén a `build(context)` metódussal valósul meg.

4.3.4. Backend tervezése és megvalósítása

A alkalmazás backendje egy mikroszolgáltatásokból álló architektúrán alapul, amely biztosítja a modularitást, a skálázhatóságot és a különböző komponensek független működését. A szolgáltatások közül kettő Node.js alapú, egyik Python nyelven készült. Így az egyes részek egységei a legmegfelelőbb technológiát használják az adott feladatra, valamint a mikroszolgáltatások egymással RESTful API-kon keresztül kommunikálnak. A fényképek tárolására a Minio-t használjuk és a titkos információkat egy vaultwarden self-hosted tárolóba mentettük.

User-management

A User-management egy különálló mikroszolgáltatásként valósult meg, Node.js és Express technológiákra építve. Ennek a szolgáltatásnak a fő feladata a felhasználók hitelesítése és azonosítása, amelyhez egy külső autentikációs szolgáltatót használunk az

OAuth2 segítségével. A bejelentkezési folyamat során a felhasználót átirányítjuk a külső hitelesítő szolgáltató oldalára, ahol a sikeres azonosítás után visszakerül az alkalmazásba.

A mikroszolgáltatás egy külön adatbázist használ, amely a felhasználókhöz tartozó adatokat, valamint a survey-kből származó kapcsolódó információkat tárolja.

Nutrition-management

A Nutrition-management mikroszolgáltatás felelős az étkezésekhez és receptekhez kapcsolódó adatok kezeléséért, beleértve az összetevőket, tápértékeket és az étkezések kalória-, szénhidrát-, fehérje-, és zsír értékeit. Ez a mikroszolgáltatás Node.js és Express alapú, és API végpontokon keresztül biztosítja a funkciókat.

A backend logikája az összetevők és receptek tárolására és kezelésére van felépítve. A tápérték számítások itt történnek.

Image-processor

Az image-processor mikroszolgáltatás egy mesterséges intelligencia alapú predikciós rendszer, amely Python és Flask használatával készült. Ez a szolgáltatás egy előre betanított modellt használ, amelyhez saját API hozzáférésünk van.

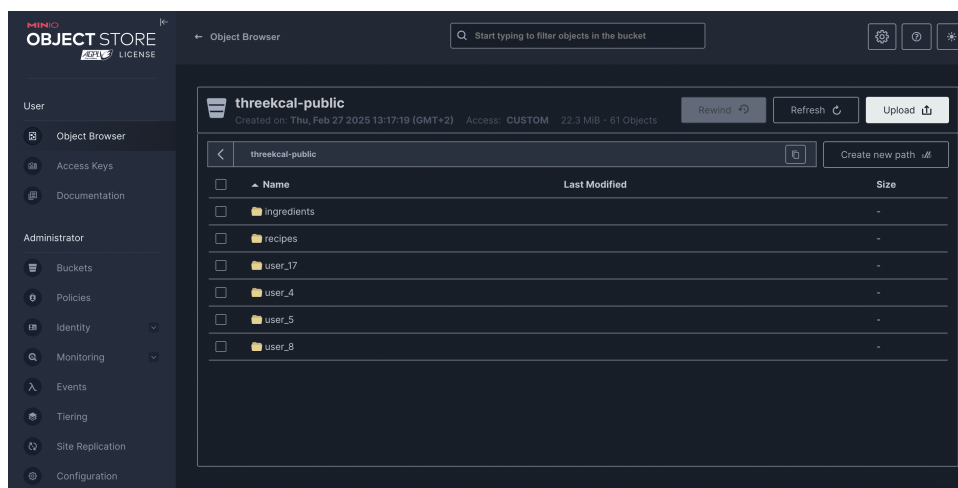
MinIO

A MinIO egy elosztott objektumtároló rendszer, amely teljes mértékben kompatibilis az Amazon S3 API-val. Az alkalmazásban a MinIO-t a képek és egyéb statikus fájlok biztonságos és hatékony tárolására használjuk.

A rendszer **bucketekre** épül, amelyekhez egyedi jogosultságokat és szabályokat állítottunk be annak érdekében, hogy kizárólag az arra jogosult komponensek férhessenek hozzá a feltöltési és törlési műveletekhez. Az adatok strukturált tárolása érdekében a következő mappastruktúrát alakítottuk ki:

- **Hozzávalók képei:** Az élelmiszer-adatbázisban szereplő hozzávalókhoz tartozó képek.
- **Receptek képei:** Az alkalmazásban megtalálható receptek illusztrációi.
- **Felhasználói képek:** Minden felhasználónak saját, elkülönített tárhelye van az étkezések- és egyéb feltöltött képei számára.

Ezek a strukturált jogosultságkezeléssel kombinálva biztosítják a rendszer adatbiztonságát és skálázhatóságát.

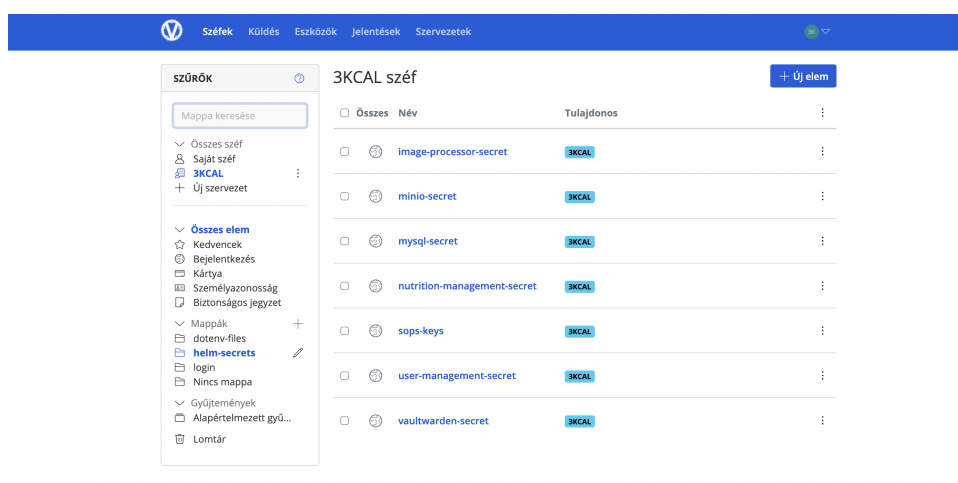


4.7. ábra. MinIO tárolási architektúra és jogosultságkezelés

Vaultwarden

A Vaultwarden egy könnyűsúlyú, önállóan üzemeltethető jelszó- és titkosadat-kezelő, amely a **Bitwarden** nyílt forráskódú alternatívájaként szolgál. Az alkalmazás fejlesztése és üzemeltetése során a Vaultwarden-t használjuk az érzékeny információk – például környezeti változók (.env fájlok), Helm-secretek és egyéb bizalmas adatok – biztonságos tárolására és kezelésére.

A titkos adatok egy központi, biztonságos széfben (*vault*) vannak tárolva, amelyhez a csapat tagjai megfelelő jogosultságokkal férhetnek hozzá. Ez a megoldás biztosítja, hogy az érzékeny adatok ne kerüljenek be a verziókezelő rendszerbe, csapattagok egyetlen helyen, könnyen és gyorsan hozzáférhessenek a szükséges titkos adatokhoz, valamint, hogy minden tárolt adat titkosítva legyen, így illetéktelen hozzáférés esetén is védve maradjon.



4.8. ábra. Vaultwarden rendszerfelépítése és biztonsági modellje

4.3.5. Web-scraping

A projekt során egy web-scraping komponenst is implementáltunk, amelynek célja receptek gyűjtése és feldolgozása volt a <https://www.delish.com/> weboldalról. A megvalósítás Python nyelven történt, a BeautifulSoup könyvtár segítségével.

A scraper automatikusan letölti az oldalon található receptoldalakat, majd ezekből kinyeri a releváns információkat, például a recept nevét, képét, hozzávalóit, elkészítési lépéseit, valamint egyéb leíró adatokat. A nyers adatok feldolgozása során különös figyelmet fordítottunk az adattisztításra, annak érdekében, hogy az információk konzisztens és strukturált formában kerüljenek további feldolgozásra.

A megtisztított és feldolgozott adatok végül a *Nutrition-management* modulhoz tartozó adatbázisba kerülnek eltárolásra. A képek tárolásához a Python beépített MinIO kliensét használtuk, amelyen keresztül a képek a saját MinIO-alapú objektumtároló szolgáltatásunkba kerülnek feltöltésre.

4.3.6. Adatbázisok

Az alkalmazás adatbázisainak tervezése során két fő adatbázist hoztunk létre, amelyek különböző funkciókat szolgálnak ki: a User-Management adatbázist és a Nutrition-Management adatbázist. A rendszer célja, hogy hatékonyan kezelje a felhasználói adatokat, valamint a táplálkozással kapcsolatos információkat, és mindkét adatbázis alapvető szerepet játszik az alkalmazás működésében. Ezek mellett, a vaulwarden is rendelkezik egy saját adatbázissal, ahova a menedzseléséhez szükséges információkat menti.

User-Management adatbázis

User	
PK	id
	google_id
	name
	email
	creation_date
	modification_date
	profile_picture
	goal
	age
	height
	weight
	gender
	lifestyle

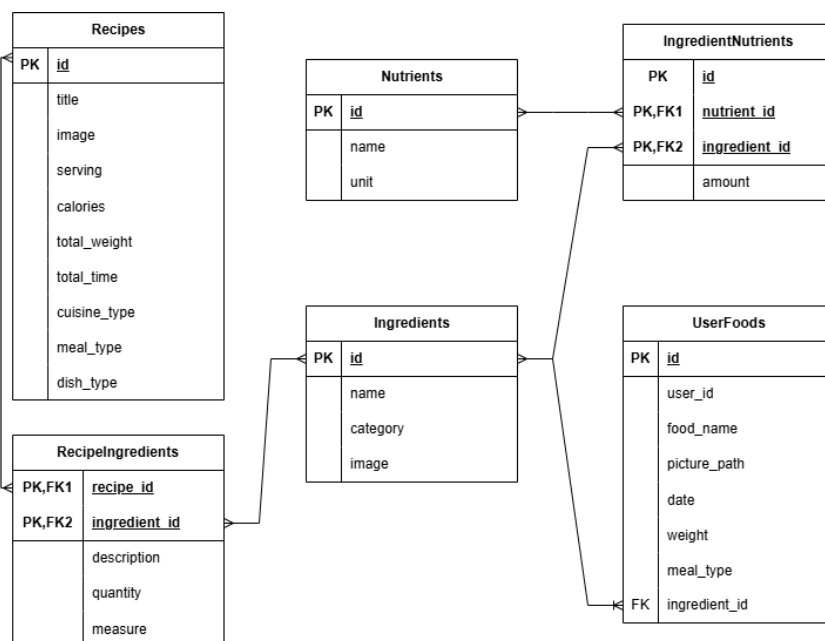
4.9. ábra. User-management adatbázis diagram

A User-Management adatbázis a felhasználók alapvető adatainak kezelésére szolgál. Ebben a rendszerbe kerülnek a felhasználói regisztrációs adatok, mint a felhasználónév, e-mail cím és jelszó, valamint a felhasználók preferenciái, mint például a napi aktivitási

szint, célok, magasság, súly, és egyéb személyes információk, amelyek a felhasználó által kitöltött kérdőívekből származnak. Mindezek az adatok együttesen biztosítják, hogy a rendszer pontosan és hatékonyan tudja kiszolgálni a felhasználó egyéni igényeit.

Az adatbázis tervezését a code-first megközelítés alapján végeztük, ami azt jelenti, hogy a kód elsődleges szerepet kapott az adatbázis struktúrájának kialakításában. A szükséges táblát, mezőket először a kódban definiáltuk, és ezek alapján hoztuk létre az adatbázist.

Nutrition-Management adatbázis



4.10. ábra. Nutrition-management adatbázis diagram

A Nutrition-Management adatbázis az étkezésekkel kapcsolatos információk tárolására szolgál. Az étkezések tápanyagtartalmát, kalóriákat, zsírokat, szénhidrátokat, fehérjéket és egyéb összetevőket tartalmazza, valamint az étkezések kategorizálását is elvégzi (pl. reggeli, ebéd, vacsora). Ezen információk alapján az alkalmazás a felhasználók napi tápanyag szükségletét és étkezési terveit tudja kiszámítani.

Az adatbázis database-first megközelítéssel lett tervezve, ami azt jelenti, hogy a táblák és azok kapcsolatai először az adatbázisban lettek létrehozva. A táblák feltöltése API-ból lekért adatokkal és datasetek összefésülésével történt, lehetővé téve a dinamikus adatkezelést és a rendszer folyamatos frissülését.

Vaultwarden adatbázis

A Vaultwarden egy MySQL adatbázist használ az adatok tárolására, amelyet maga a rendszer hoz létre és kezel. Ez az adatbázis nem általunk tervezett vagy testreszabott, hanem a Vaultwarden működéséhez elengedhetetlen komponens, amely biztosítja a szükséges adatstruktúrát és tárolási lehetőségeket.

4.4. Kitelepítés

A cég biztosított számunkra egy dedikált Kubernetes névteret a vállalati klaszteren belül, amelybe a mikroszervizeinket és az adatbázisainkat telepítettük. Kezdetben Deployment-manifesztekkel végeztük a telepítést, amelynek során StatefulSet erőforrásokat is létrehoztunk az állapotot megőrző komponensekhez. Később a telepítési folyamatot Helm chartok segítségével automatizáltuk és menedzseltük.

Cert-Manager

A Cert-Manager egy Kubernetes-integráció, amely az SSL/TLS tanúsítványok kezeléséért felelős. A rendszer ezen komponense a Let's Encrypt szolgáltatást alkalmazza a tanúsítványok igénylésére és megújítására, biztosítva ezzel a hálózati kommunikáció titkosítását és biztonságát az alkalmazás és a felhasználók között.

Traefik

A Traefik terheléelosztó, amely a Kubernetes szolgáltatásokat kezeli. A komponens feladata az összes bejövő külső kérés kezelése, és annak megfelelő irányítása a belső szolgáltatások felé. A Traefik külön namespace-ben, a traefik-ns környezetben működik, amely biztosítja a megfelelő izolációt és konfigurálhatóságot.

Helm chartok

A Helm egy csomagkezelő eszköz Kuberneteshez, amely lehetővé teszi az alkalmazások egyszerű telepítését, frissítését és kezelését. A Helm chart egy olyan konfigurációs csomag, amely tartalmazza az adott alkalmazás telepítéséhez szükséges Kubernetes-manifeszteteket, például a Deployment, Service, ConfigMap és Secret fájlokat. Projektünkben különböző forrásokból származó Helm chartokat használunk az egyes szolgáltatásokhoz, amelyeket az **Artifact Hub**-on keresztül is elérhetünk. Az [Artifact Hub](#) egy központi tároló, amely lehetővé teszi a Helm chartok, operátorok és egyéb Kubernetes-csomagok felfedezését és kezelését.

A MinIO és MySQL esetében a Bitnami által karbantartott, jól dokumentált Helm chartokat alkalmazzuk, amelyek megbízható és könnyen konfigurálható megoldást nyújtanak. Ezeket a chartokat az Artifact Hub-on keresztül érhetjük el, ami biztosítja, hogy mindig a legfrissebb és stabil verziókat használjuk. A Vaultwarden és a Bittorrent-kliensek telepítéséhez a bittorrent-helm repository megfelelő chartjait alkalmazzuk, amelyek szintén elérhetők az Artifact Hub-on.

A többi szolgáltatás esetében saját Helm chartokat hoztunk létre, figyelembe véve az adott alkalmazás egyedi igényeit, és megírva a szükséges YAML fájlokat. Ez lehetővé teszi számunkra, hogy teljes mértékben testreszabott és rugalmas megoldásokat biztosítsunk, miközben a meglévő, közösségi chartok előnyeit is kihasználjuk.

Az általunk alkalmazott megközelítés biztosítja, hogy a lehető legjobban ötvözzük a közösség által karbantartott chartok nyújtotta stabilitást és a saját fejlesztésű chartok rugalmasságát, így optimalizálva a Kubernetes-alapú környezetünk működését.

Helm Secrets Plugin

A **Helm Secrets Plugin** egy hasznos eszköz a Kubernetes környezetekben való titkosítás kezelésére. Segítségével titkos adatokat, például API kulcsokat, jelszavakat és egyéb érzékeny információkat titkosíthatunk a Helm chartokban, miközben megőrizzük a konfigurációs fájlok biztonságát. A plugin a **sops (Secrets Operations)** eszközt használja a titkos fájlok kezelésére, amely lehetővé teszi a titkosított fájlok egyszerű és biztonságos kezelését anélkül, hogy azok kikerülnének a verziókezelő rendszerbe.

A projektünkben a **SOPS** alkalmazását választottuk a titkosítási folyamatokhoz, és az **AGE** (Actually Good Encryption) titkosítási rendszert használjuk. Az AGE egy modern, egyszerűsített titkosítási eszköz, amely **publikus kulcsú titkosítást** alkalmaz. A titkosításhoz a nyilvános kulcsot alkalmazzuk, míg a titkos fájlok visszafejtéséhez a **privát kulcs** szükséges. Az AGE-t használjuk a titkos adatok titkosítására, és a **Helm Secrets Plugin** az alkalmazott privát kulcsot használja a titkosított fájlok visszafejtésére, biztosítva, hogy csak azok az entitások férjenek hozzá a titkos adatokhoz, akik rendelkeznek a megfelelő privát kulccsal.

```
user-management.enc.yaml 1.50 KIB Edit Replace Delete

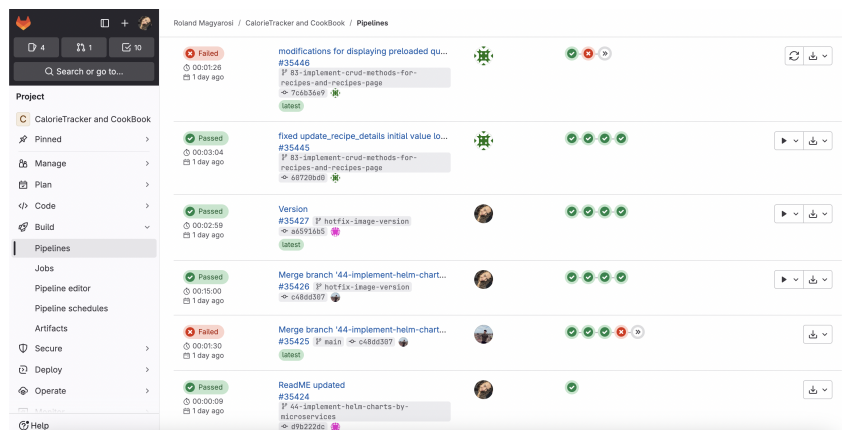
1 secret:
2   data:
3     GOOGLE_CLIENT_SECRET: ENC[AES256_GCM,data:qhJDMdV20Mvh7U80LZsryTkQ+78ET+e1VTX66Ag2HjCjJ9c=,iv:Jn1CLL40Jt8hKthPEvomQ8/MSgpN
4     SECRET: ENC[AES256_GCM,data:Yq+Ivtpe1pvu3Zj6,iv:3Rr+JRxgwKdeuM82paLxKqdc+UEZIA0LzqNHvvIvYw=,tag:qQvur0mWScBy78TQ60Lw==,
5     MYSQL_PASSWORD: ENC[AES256_GCM,data:SbL+YndobqBuLbhh+QU,iv:orjCRuTFVujtJfMBBvgH4S9dJEFKVHus98ABZ8fvsNw=,tag:EzEaWh63maig4
6
7 sops:
8   kms: []
9   gcp_kms: []
10  azure_kv: []
11  hc_vault: []
12  age:
13    - recipient: age1qdvkrte6mvcvtv3rh2huy2kze4emcpvdzahg@rpns7etgcreagx5y4w53k
14      enc: |
15        -----BEGIN AGE ENCRYPTED FILE-----
16        YW9LLWVvY3J5cHRpb24ub3JhL3YxCl0+1FgyNTUxOjAsAeFFjdhQh8F0ME9W0VNVK
17        N1dqT196amJxTDA5NWw1d0p2H3dQSGNidLV3C1t1SNWVWLB1WTUzWfEa0R0b0Js
18        WKEyOFLNb0p1Umsvb3BHTWZTN0xpam8KLS0tIDZ1cVRCaEkuNEZFZUx3Y0Y2dVQv
19        YkY1Z1JvMCIid1V0NHQ3T1I4OTg4VTAk7K2KYSSg96l29hw2s4ZF5SqUaVpHPuK
20        A+yAX4z16pmL570a6Du3k6j9UsKslamyYKShmqZwq6vn0H3DQxKcg==
21        -----END AGE ENCRYPTED FILE-----
22  lastmodified: "2025-03-20T16:47:48Z"
23  mac: ENC[AES256_GCM,data:5/SzR8JjgKKdHTgkLaBhM/Q8xLc0GDS20A3I8z1wbdVIRNPF1J0+u0MRkE9kMT68xRt09h1cHfV6nhHs5YfNekC640ax7enn0xfo
24  pgp: []
25  unencrypted_suffix: _unencrypted
26  version: 3.9.4
```

4.11. ábra. Titkosított helm secret

4.4.1. CI-Pipeline

A CI (Continuous Integration) pipeline egy automatizált folyamat, amely biztosítja a forráskód folyamatos ellenőrzését és tesztelését, még azelőtt, hogy a fejlesztések éles környezetbe kerüljenek. A pipeline több egymásra épülő szakaszból (stage-ből) áll, amelyek a következő fő lépéseket tartalmazzák.

A pipeline csak akkor indul el, egy bizonyos komponensre, ha változás történt a szolgáltatások (service-ek) vagy a frontend kódbázisában. Továbbá, a konténerkép (image) csak akkor pusholódik ki a registry-be, ha az adott változtatás egy merge request (MR) részeként történik. A pipeline push esemény esetén is triggerelődik, biztosítva a folyamatos integrációt.



4.12. ábra. Pipeline

Init Stage

Az init pipeline egy alapértelmezett szakasz, amely mindig elindul. Ebben a szakaszban előkészítjük a környezetet a további lépésekhez.

Coding Standards Stage

Ebben a fázisban ellenőrizzük a kódminőséget és a formázási szabályokat. Az alábbi lépések zajlanak le:

- A szükséges csomagok és függőségek telepítése.
- A forráskód buildelése a mobile rész esetén.
- A kód lintelése, azaz formázási és szintaktikai hibák keresése.

Test Stage

Ebben a szakaszban a rendszer automatikus teszteket futtat a kód minőségének biztosítása érdekében. A cél az, hogy az esetleges hibákat már fejlesztési szakaszban észleljük, mielőtt azok bekerülnének a végleges verzióba.

Check Stage

A check szakasz feladata annak ellenőrzése, hogy a buildelendő konténerkép már létezik-e a GitLab Container Registry-ben. Ezzel elkerülhető a felesleges újraépítés, ha a kép már létezik az adott verzióval.

Docker Stage

A Docker szakasz a konténerkép létrehozását és regisztrációját végzi. Két fő lépésből áll:

- Egyszerű build folyamat, amely létrehozza a szükséges konténerképet.
- Build and push folyamat egy külön címkével (tag), amely feltölti az újonnan létrehozott képet a GitLab Container Registry-be, de kizárólag akkor, ha egy merge request esetén történik a változtatás.

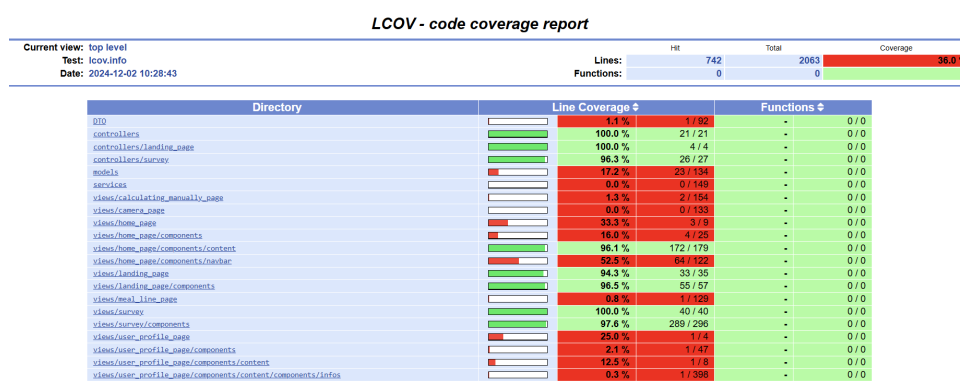
A konténerkép neve a szolgáltatás nevéből (service name) és a verziószámból áll össze, biztosítva az egyértelmű verziókezelést és azonosíthatóságot a regisztrált képek között.

A kód szementálására és elkülönítésére ASCII-ART-ot használtunk, ami színesebbé és átláthatóbbá tette a kódot.

4.5. Tesztelhetőség

A tesztelhetőség kiemelt szerepet kapott az alkalmazás fejlesztése során, mivel alapvetően hozzájárul a kódminőség fenntartásához és a jövőbeli fejlesztések zökkenőmentességéhez. Az automatizált tesztelés és a hibaellenőrzés olyan megoldásokat nyújt, amelyek segítenek az alkalmazás stabilitásának biztosításában és a hibák gyors azonosításában.

Az automatizált tesztelés az alkalmazás összes komponensében – Flutter frontend, Node.js backend és Python képfeldolgozó backend – átfogóan megvalósult. A Flutter környezetben unit- és widget tesztek készültek, amelyek az alkalmazás logikai és felhasználói felületének különböző komponenseit fedik le, mint például a landoló oldal, a főoldal, a profil oldal, a kérdőív, valamint a kamera oldal. A unit tesztek a controllerek és modellek helyes működését ellenőrzik, míg a widget tesztek a UI elemek interaktivitását tesztelik. A tesztekhez mock objektumokat használtunk a külső függőségek szimulálására, ezzel biztosítva a tesztelt kódrészek izolálását.



4.15. ábra. Flutter coverage

A Node.js backend tesztelését a Jest tesztkeretrendszerrel végeztük. Itt külön figyelmet fordítottunk a hitelesítési folyamatok, például a bejelentkezés és az OAuth integrációk tesztelésére. A middleware-ek és külső függőségek, mint az adatbázis és az Express-session, mockolása lehetővé tette az egyes végpontok különálló tesztelését. Az alapvető funkciók, például a helyes és helytelen bejelentkezési kísérletek, kötelező mezők hiánya, valamint a hitelesítési hibaágak is részletes vizsgálaton estek át.

A Python képfeldolgozó backendnél a pytest keretrendszert alkalmaztuk. Az egysegtesztek biztosították a képfeldolgozó algoritmus helyes működését, míg az integrációs tesztek az inferencia szolgáltatásokkal való kommunikációt ellenőrizték. Kiemelt figyelmet fordítottunk a hibakezelésre, például érvénytelen képadatok vagy hiányzó bemenet esetén.

A projekt sikerének alapvető feltétele a kód minőségének fenntartása és a tesztek széleskörű lefedettsége. Ehhez különböző eszközöket és konfigurációkat vezettünk be mindhárom fejlesztési környezetben. A Python backendhez több eszköz biztosítja a kódstílus és a hibamentesség fenntartását, ilyen például a pylint, amely az egyik legfontosabb eszköz a statikus kódelemzéshez, ugyanis segít az általános hibák, stílusbeli eltérések és a nem hatékony megoldások azonosításában. Az alkalmazásunk konfigurációjában a maximális sorhossz 120 karakter, valamint bizonyos figyelmeztetéseket (pl. dokumentáció hiánya) letiltottunk, hogy a fejlesztők a lényegi kérdésekre koncentrálhassanak. Továbbá hasznél-

tuk a black és az isort eszközöket is, amelyek biztosítják a kód egységes formázását és az importok megfelelő sorrendjét, szintén 120 karakteres sorhossz mellett. Ami a tesztfedettségrel illeti, a pytest-et a coverage eszköz kombinációjával futtatjuk le, ami segít abban, hogy a tesztek lefedettsége pontosan mérhető legyen. A taskipy használatával egyszerűen futtathatók ezek az eszközök egyetlen parancs segítségével:

```
task test = "coverage run --source=src -m pytest && coverage report"
```

Name	Stmts	Miss	Cover

src\DTO\food_dto.py	7	0	100%
src\image_processor.py	18	9	50%
src\process_image.py	37	1	97%

TOTAL	62	10	84%

4.16. ábra. Python képfelismerő backend teszt lefedettsége

A Node.js backend fejlesztését az ESLint segíti, amely hatékony lintelési szabályokat és konfigurációkat biztosít a JavaScript kódhoz. Az eslint.config.mjs fájl különböző környezetekhez definiál globális változókat (a Node.js környezethez a globals.node, a böngésző környezethez a globals.browser és a Jest tesztkörnyezethez a globals.jest), valamint az ajánlott szabályokat a @eslint/js csomag biztosítja, amely garantálja a legjobb gyakorlatok alkalmazását.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	58.64	52.5	24	59	
src	85	91.66	50	87.17	
app.js	85	91.66	50	87.17	30,40-41,70-74
src/controllers	47.88	41.66	22.22	47.88	
authController.js	26.66	0	0	26.66	17,21-26,31-40,45-48,53-75,80-88
UserController.js	84.61	62.5	100	84.61	16-17,44-45
src/middlewares	25	0	0	25	
authMiddleware.js	25	0	0	25	2-5
src/routes	100	100	100	100	
authRoutes.js	100	100	100	100	
userRoutes.js	100	100	100	100	
src/services	30	0	0	30	
authService.js	31.81	100	0	31.81	7-11,15-44
userService.js	25	0	0	25	4-28
Test Suites: 3 passed, 3 total Tests: 12 passed, 12 total Snapshots: 0 total Time: 1.232 s Ran all test suites.					
File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	83.45	63.33	61.53	83.45	
nutrition-management	84.52	67.85	70	84.52	
app.js	84.52	67.85	70	84.52	37,54,118-119,153-154,187-188,193,197-199,204
nutrition-management/src/config	100	100	100	100	
database.js	100	100	100	100	
nutrition-management/src/middleware	25	0	0	25	
authentication.js	25	0	0	25	6-24
nutrition-management/src/models	97.5	100	50	97.5	
index.js	94.44	100	50	94.44	41
ingredient.js	100	100	100	100	
ingredientNutrient.js	100	100	100	100	
nutrient.js	100	100	100	100	
recipe.js	100	100	100	100	
userFood.js	100	100	100	100	
Test Suites: 6 passed, 6 total Tests: 11 passed, 11 total Snapshots: 0 total Time: 1.835 s Ran all test suites.					

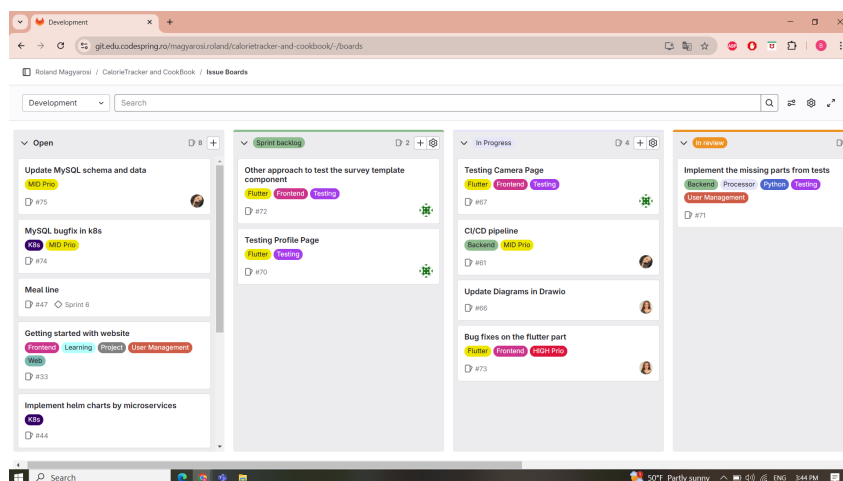
4.17. ábra. Node.js felhasználókezelő és nutríció kezelő backend-ek teszt lefedettségei

A Flutter frontendben az analysis_options.yaml fájl biztosítja a kódstílus és a minőség fenntartását, valamint a Flutter-specifikus ajánlásokat. Az alapértelmezett Flutter linter szabályok, mint például a prefer_const_constructors (az állandó konstruktorok használata, ami javítja a teljesítményt), az unused_import (a nem használt importok eltávolítása) és a prefer_final_fields (a final mezők használata, ahol lehetséges, a kód stabilitása érdekében) olyan területeket fednek le, mint a kódstílus, a teljesítményoptimalizáció és a null safety. A tesztlefedettséget a flutter test --coverage parancs méri, amely vizuális elemzést is lehetővé tesz az lcov eszközzel.

4.6. GitLab Issue Board

A GitLab Issue Board lehetővé teszi a feladatok vizuális nyomon követését és hatékony kezelését. A GitLab Issue Board segítségével a csapatok könnyen követhetik az egyes feladatok állapotát, rendkívül hasznos a projekt menedzselése, mivel a csapat tagjai láthatják, hogy mely feladatok vannak folyamatban, melyek készültek el, és melyek várnak még megoldásra.

Az Issue Board alapvetően az Issue-kat (feladatokat) táblázatos formában jeleníti meg, ahol az egyes kártyák a különböző feladatokat reprezentálják. A kártyák különböző



4.18. ábra. Gitlab Issue board

oszlopokba rendezhetők, például: Open, Print Backlog, In Progress, In Review, Done és Closed, ami segít a csapat számára az átláthatóságban és a feladatok állapotának nyomon követésében.

A projekt során alkalmazott fejlesztési metodológia biztosította, hogy a csapat gyorsan alkalmazkodjon a változó igényekhez és folyamatosan fejlődjön. A munkafolyamatokat heti sprintekben szerveztük meg, amelyek lehetővé tették, hogy minden sprint végén értékeljük az elvégzett munkát, és a csapat visszajelzései alapján finomhangoljuk a következő ciklusokat. A heti sprintek segítettek a folyamatos előrehaladás biztosításában, miközben lehetőséget adtak arra, hogy a csapat gyorsan reagáljon a felmerülő problémákra és új prioritásokra.

A scrum poker alkalmazása hozzájárult ahhoz, hogy a csapattagok közösen, becsüljék meg a feladatok bonyolultságát és időigényességét, story point-otkat rendelve minden feladathoz. Ez a technika nemcsak a feladatok tervezésében segített, hanem elősegítette a csapat közötti kommunikációt és a közös megértést a projekt céljait illetően.

A label használat a projekt nyomon követésére és kategorizálására szolgált. A label rendszer segített a feladatok prioritizálásában és a munkafolyamatok gyors átlátásában, biztosítva, hogy minden csapattag tisztában legyen a teendőikkel és azok állapotával.

Végül, a retrospektív megbeszélések minden sprint végén lehetőséget biztosítottak arra, hogy a csapat értékelje a végzett munkát, és meghatározza a javítási lehetőségeket. Ez segített a csapatot abban, hogy folyamatosan finomítsa a munkafolyamatokat, növelve a hatékonyságot és biztosítva a projekt sikeres előrehaladását.

Ezek a gyakorlatok összességében hozzájárultak a projekt átláthatóságához, a csapat hatékony munkavégzéséhez és a felmerülő problémák gyors megoldásához, biztosítva, hogy a fejlesztési folyamat dinamikusan és rugalmasan reagáljon a változó körülményekre.

5. fejezet

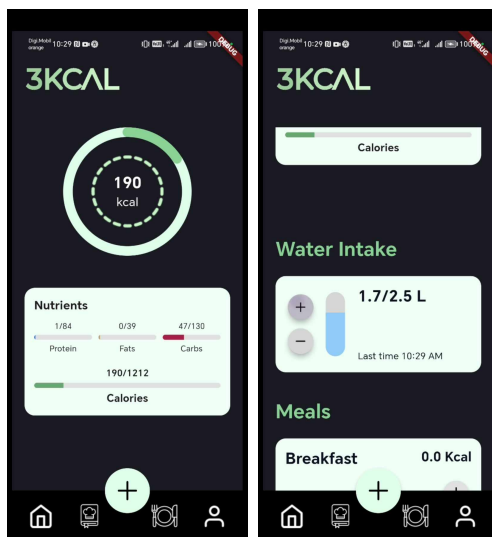
A rendszer bemutatása

Egy kalóriaszámláló és követő alkalmazás célja az, hogy segítse a felhasználókat a napi kalóriabevitelük és energiafelhasználásuk nyomon követésében.

5.1. Fő funkciók

5.1.1. Bevitt tápanyagértékek és vízfogyasztás naplózása

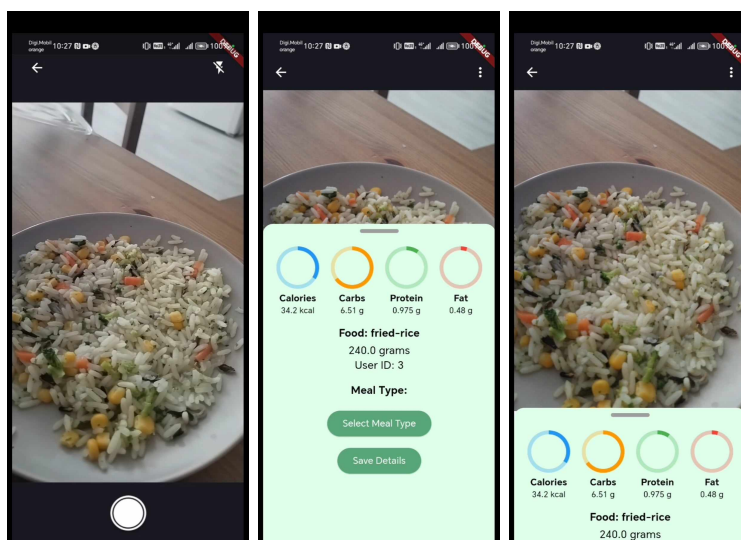
A kalóriaszámláló alkalmazások egyik alapvető funkciója a napi tápanyagbevitel részletes nyomon követése. A felhasználók rögzíthetik az elfogyasztott ételeket és italokat, amelyek tápanyagértékeit (kalória, fehérje, zsír, szénhidrát, vitaminok) az alkalmazás automatikusan kiszámítja. Ezen kívül lehetőség van a napi vízfogyasztás naplózására is, amely segíti a megfelelő hidratáció fenntartását. Az adatok elemzése alapján az alkalmazás személyre szabott tanácsokat nyújthat az egészségesebb táplálkozás érdekében.



5.1. ábra. Main Screen

5.1.2. Kép alapú tápanyagérték felismerés

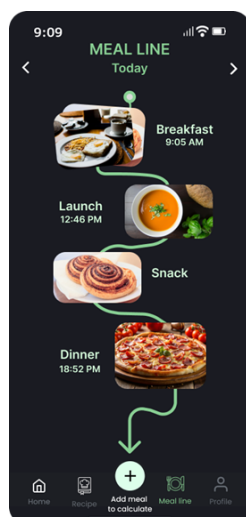
A kép alapú tápanyagérték felismerés a kalóriaszámláló alkalmazásokban egy új, innovatív funkció, amely lehetővé teszi az ételek tápanyagértékeinek meghatározását csupán a termék vagy étel fényképének segítségével. Az alkalmazások mesterséges intelligencia (MI) algoritmusokat használnak, például képfelismerő rendszereket, hogy az ételt automatikusan azonosítsák, és a hozzá tartozó tápanyagértékeket az adatbázisból lekérjék. Ez a módszer jelentősen gyorsítja a tápanyagbevitel rögzítését, mivel a felhasználóknak nem kell manuálisan beírniuk az ételt vagy vonalkódot beolvasniuk. A képfelismerés folyamata folyamatosan fejlődik, és a pontosabb adatgyűjtés érdekében egyre inkább képes a komplex ételek és étkezések felismerésére is. Illetve itt lehetősége van a felhasználónak kiválasztani, hogy melyik ételtípushoz tartozik az elfogyasztani kívánt étel: reggeli, ebéd, vacsora, snack. Amikor kiválasztotta, automatikusan hozzáadódik az érték a main screenen levő vizualizáló komponensekhez, illetve megjelenik, hogy melyik ételtípusnál mennyi kalóriát vittünk be.



5.2. ábra. Étél felismerés, és annak tápanyagértékeinek megjelenítése

5.1.3. Meal Line - Elfogyasztott ételek nyomon vizuális követése

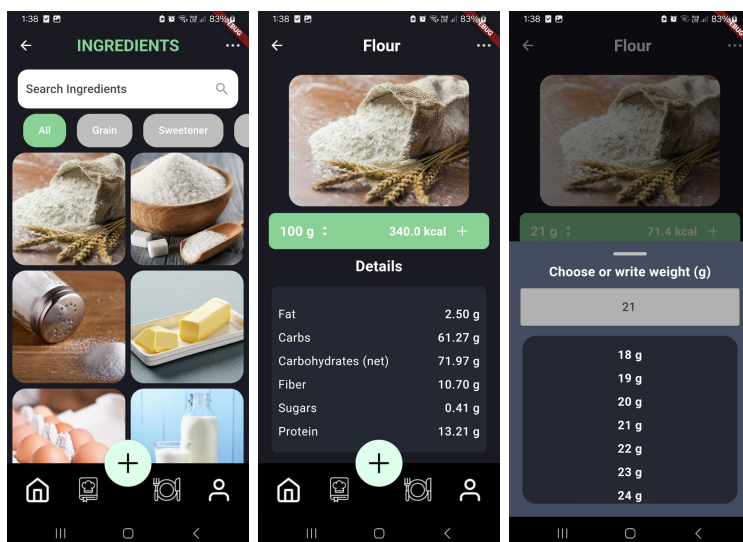
A "Meal Line" funkció a kalóriaszámláló alkalmazásokban egy vizuális eszközként szolgál az elfogyasztott ételek nyomon követésére. A felhasználók naponta rögzíthetik az étkezéseiket, amelyek időpont szerint rendezve és képekkel illusztrálva jelennek meg a naplóban. A Meal Line célja, hogy egy grafikus idővonalat biztosítson, amely segít az étkezések és a tápanyagbevitel könnyebb áttekintésében. A vizuális elemek (például ételfotók, étkezések időpontjai és tápanyagok) segítségével a felhasználó könnyen figyelemmel kísérheti, hogy milyen típusú ételek kerültek a napi étkezésekbe, és hogy ezek hogyan illeszkednek az egészségügyi céljaikhoz. A *Meal Line* motiváló eszközként is szolgálhat, mivel egy vizuális megjelenítést ad a felhasználó előrehaladásáról.



5.3. ábra. Meal Line

5.1.4. Manuális tápanyagbevitel

A manuális tápanyagbevitel lehetősége a kalóriaszámláló alkalmazásokban lehetővé teszi a felhasználók számára, hogy saját maguk rögzítsék a napi étkezéseiket és az azokhoz tartozó tápanyagértékeket. Ez különösen hasznos, amikor az alkalmazás adatbázisában nem található meg az étel vagy ital, amit a felhasználó fogyasztott. A manuális bevitel során a felhasználók megadhatják az étel nevét, a mennyiségét, és az alkalmazás ez alapján kiszámítja a kalóriát és a makrotápanyagokat (fehérje, zsír, szénhidrát). Ezen kívül lehetőség van egyedi tápanyagértékek hozzáadására is, például a felhasználó által készített ételek esetén. A manuális bevitel lehetőséget ad arra, hogy a felhasználó teljes mértékben ellenőrizze az étkezéseit, és biztosítja, hogy minden étkezés pontosan rögzítésre kerüljön.



5.4. ábra. Alapanyagok manuális hozzáadása

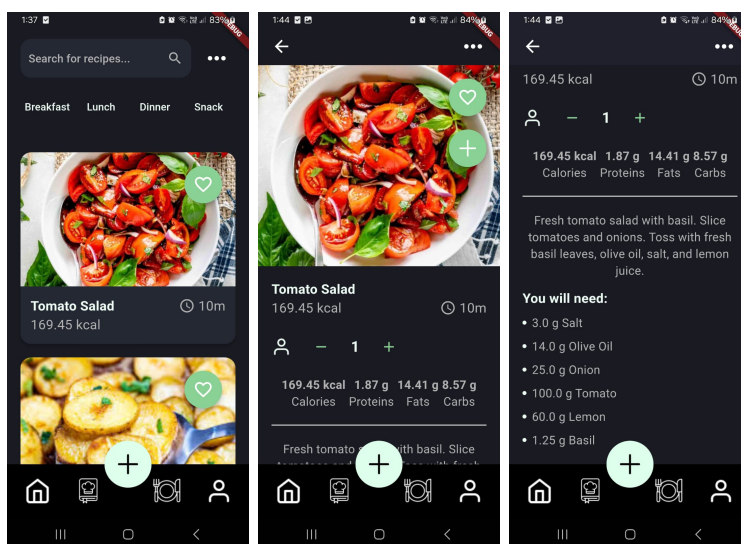
Újdonságként a rendszer lehetőséget biztosít arra is, hogy a felhasználó új alapanyagokat (ingredienteket) adjon hozzá az adatbázishoz. Az újonnan hozzáadott alapanyagok

kat az Ollama AI ellenőrzi, hogy azok létező, valós alapanyagok-e, és hogy megfelelőek-e az adatbázisba történő felvételre. A felhasználóknak lehetőségük van saját alapanyagaik hozzáadására, frissítésére és törlésére is, így személyre szabhatják a rendszerben tárolt ételmenülistát a saját étrendjükhöz igazodva.

5.1.5. Receptek kezelése - CookBook

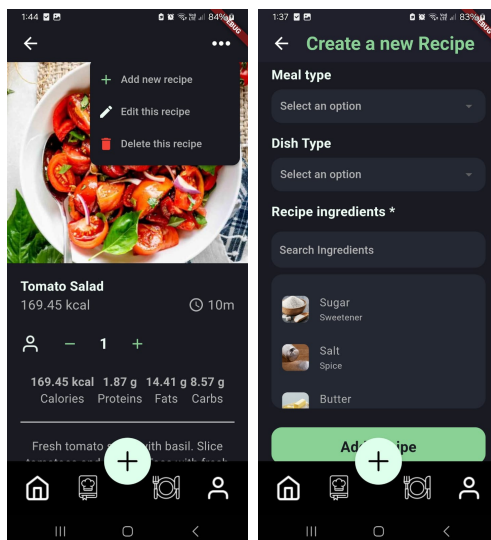
Az alkalmazás egy saját receptadatbázissal rendelkezik, amelyet webes forrásokból történő adatgyűjtés (scraping) segítségével állítunk össze. Ez lehetővé teszi, hogy a felhasználók változatos és naprakész receptgyűjteményből válogathassanak, amely folyamatosan bővül és frissül.

A receptadatbázis tartalmazza az egyes ételek részletes leírását, elkészítési módját, hozzávalóit, valamint a tápanyagértékeket. A receptek szűrhetők különböző szempontok alapján.



5.5. ábra. Alapanyagok manuális hozzáadása

A felhasználóknak lehetősége van saját receptek hozzáadására is. A beküldött receptek tartalmazhatják az elkészítési lépéseket, a hozzávalókat (amelyek akár manuálisan, akár az adatbázisból választhatók ki), valamint a tápanyag-információkat. A felhasználó ezeket a recepteket később módosíthatja vagy törölheti, így az alkalmazás teljes mértékben testre szabhatóvá válik az egyéni igények szerint.

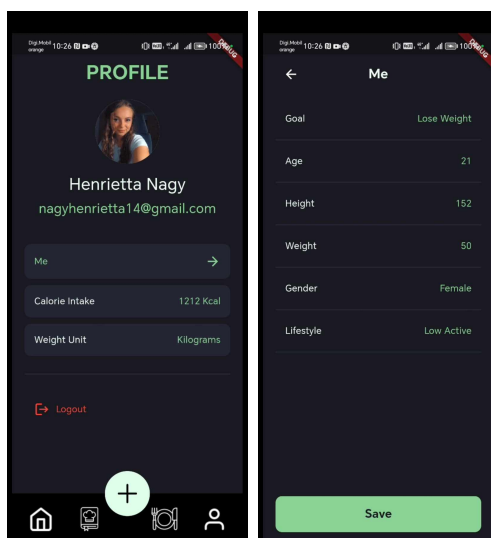


5.6. ábra. Saját recept hozzáadása

A saját receptek kezelése hozzájárul ahhoz, hogy a felhasználók személyesebb kapcsolatot alakíthassanak ki az étrendjükkel, miközben megőrzik a pontos tápanyagkövetést.

5.1.6. Profil kezelés - saját adatok módosítása

A profil kezelés funkció lehetővé teszi a felhasználók számára, hogy személyes adatokat módosítsanak a kalóriaszámláló alkalmazásban. Ezen a felületen a felhasználók frissíthetik alapvető információikat, mint például a nevüket, életkorukat, nemüket, magasságukat, súlyukat és életmódjukat. A módosítások segítenek abban, hogy az alkalmazás pontosabban számolja ki a napi kalóriabevitelt és tápanyagigényt a felhasználó aktuális állapotához és céljaihoz igazodva. A profil frissítése után az alkalmazás új, személyre szabott ajánlásokat és célokat kínál, figyelembe véve a módosított adatokat.



5.7. ábra. Profile Screen

6. fejezet

Összegzés

Összességében a dolgozat célja egy modern, modulárisan felépített, mikroszolgáltatás-alapú kalóriaszámláló alkalmazás bemutatása volt, amely hatékonyan támogatja a felhasználókat a tudatos táplálkozásban és egészséges életmód fenntartásában. Az architektúra kialakítása során kiemelt figyelmet kapott a skálázhatóság, a testreszabhatóság és a jövőbeni bővíthetőség. A funkciók – mint például a napi kalória és vízfogyasztás követése, az étkezések nyilvántartása, valamint a receptek olyan eszközöket kínálnak, amelyek megkönnyítik az egyéni célok elérését.

6.1. Kinek ajánlott?

A kalóriaszámláló és táplálkozáskövető alkalmazások széles körben hasznosak lehetnek, és különböző célcsoportok számára ajánlottak. Elsősorban azoknak, akik szeretnék jobban nyomon követni a napi tápanyagbevitelt, kalóriát és egyéb fontos tápanyagokat. Ajánlott:

- **Egészségtudatos emberek:** Akik figyelnek a táplálkozásukra, szeretnék fenntartani vagy javítani az egészségi állapotukat.
- **Fogyókúrázók:** Akik a testsúlycsökkentés érdekében szeretnék nyomon követni a kalóriabevitelüket, és segítenek az optimális étrend kialakításában.
- **Sportolók és aktív életmódot folytatók:** Akik napi szinten nyomon követik az energiabevitelüket és tápanyagokat, hogy támogassák a teljesítményüket és regenerációjukat.
- **Egészségügyi problémákkal küzdők:** Akiknek a táplálkozásuk figyelemmel kísérése szükséges a krónikus betegségekkel (pl. cukorbetegség, szívbetegségek) való kezelése során.

Ezek az alkalmazások segíthetnek az egészséges életmód fenntartásában, a tápanyagok megfelelő elosztásában és a testre szabott táplálkozási célok elérésében.

6.2. Továbbfejlesztési lehetőségek

A rendszer fejlesztése során számos ötlet és lehetőség merült fel, amit a jövőben érdemes lenne megvalósítani. Ezek közé tartozik új adathalmazok integrálása, a funkciók továbbgondolása, valamint a technikai háttér erősítése.

Snack Track adathalmaz integrálása

A Snack Track Dataset integrálása egy fontos következő lépés lehet, mivel rengeteg új alapanyagot tartalmaz, különböző boltok kínálatából, világszerte. Ez jelentősen bővítené az alkalmazás adatbázisát, és lehetővé tenné, hogy a felhasználók többféle alapanyagból válogathassanak – akár helyi, akár nemzetközi termékekről van szó. Ez nemcsak a pontosságot növelné az összetevők felismerésében, hanem a tápanyagtartalom számításában is segítené.

CookBook funkciók bővítése

A tervek között szerepel a CookBook modul funkcióinak továbbfejlesztése, hogy még pontosabban igazodjon a felhasználók étrendi igényeihez és preferenciáihoz. Ennek részeként több új szűrési lehetőség bevezetését tervezzük, például alapanyagok, kalóriatartalom vagy allergének szerinti keresést.

Emellett cél egy személyre szabott ajánlórendszer kialakítása is, amely a felhasználók korábbi aktivitása és preferenciái alapján javasolna recepteket, illetve lehetőséget adna arra is, hogy a felhasználók kiválasztott hozzávalók alapján kapjanak releváns receptajánlásokat.

CI/CD pipeline-ok továbbfejlesztése

A rendszer jelenleg folyamatos integrációval (CI) rendelkezik, de a következő lépés a folyamatos szállítás (CD) automatizálása. A cél, hogy a kódfrissítések automatikusan, manuális beavatkozás nélkül kerüljenek az üzemeltetésbe. Ezáltal a frissítések gyorsabban és zökkenőmentesebben kerülhetnek ki, miközben minimalizálják a hibák lehetőségét és növelnék a fejlesztési ciklusok sebességét.

Ábrák jegyzéke

2.1. Alkalmazások kiértékelése	18
2.2. Yazio főoldal	19
2.3. Főoldal és képfelismerés	21
2.4. SnapCalorie főoldal és adatbevitel oldal	22
3.1. Use-case diagram a felhasználói interakciókhoz	26
4.1. Architektúra diagram	34
4.2. Splash képernyő (bal oldal), első bevezető oldal (középen), második bevezető oldal (jobb oldal).	40
4.3. Cél meghatározása (kép bal oldalon), nem kiválasztása (középen), testsúly megadása (jobb oldal).	41
4.4. A főképernyő UI terve	41
4.5. Kamera megjelenítése(bal oldal), képfelismerés, illetve a felismert étel adatai(jobb oldal)	42
4.6. Home Page osztálydiagram	43
4.7. MinIO tárolási architektúra és jogosultságkezelés	46
4.8. Vaultwarden rendszerfelépítése és biztonsági modellje	46
4.9. User-management adatbázis diagram	47
4.10. Nutrition-management adatbázis diagram	48
4.11. Titkosított helm secret	51
4.12. Pipeline	51
4.13. Pipeline ASCII-ART	53
4.14. Pipeline architektúra	53
4.15. Flutter coverage	54
4.16. Python képfelismerő backend teszt lefedettsége	55
4.17. Node.js felhasználókezelő és nutrióció kezelő backend-ek teszt lefedettségei	56
4.18. Gitlab Issue board	57
5.1. Main Screen	58
5.2. Étél felismerés, és annak tápanyagértékeinek megjelenítése	59
5.3. Meal Line	60
5.4. Alapanyagok manuális hozzáadása	60
5.5. Alapanyagok manuális hozzáadása	61
5.6. Saját recept hozzáadása	62
5.7. Profile Screen	62

Táblázatok jegyzéke

3.1. Google-fiókkal történő bejelentkezés use case specifikációja	28
---	----

Irodalomjegyzék

- [Aim24] Aimojo. Yazio review, pricing, features and alternatives - sep 2024, 2024. Accessed: 2024-11-24.
- [App24] Appsrnk. Yazio review: Is it worth the download?, 2024. Accessed: 2024-11-24.
- [cal24] Calorie mama ai: Diet counter. *AppsHunter*, 2024.
- [Car17] Chris Carr. Never count calories again with calorie mama. *NewsReports*, 2017.
- [FO23] FAO and World Health Organization. Macronutrient recommendations, 2023. Accessed: 2024-11-24.
- [Kea17] Lauren Keating. Calorie mama is the nutrition app that uses photo recognition and ai to track food. *Tech Times*, 2017.
- [Kes23] Esra Özey Keskin, Bülent; Köse. The effect of calorie tracking on eating behaviors of university students. *Journal of Science Learning*, 2023.
- [Org23a] World Health Organization. Alcohol and nutrition, 2023. Accessed: 2024-11-24.
- [Org23b] World Health Organization. Establishing global nutrient requirements, 2023. Accessed: 2024-11-24.
- [Org23c] World Health Organization. Healthy diet, 2023. Accessed: 2024-11-24.
- [Org23d] World Health Organization. Nutrition, 2023. Accessed: 2024-11-24.
- [Org23e] World Health Organization. Obesity and overweight, 2023. Accessed: 2024-11-24.
- [Spr23] Springer. Calorie tracking and its benefits. *SpringerLink*, 2023.
- [SS22] Samsun Naher Muhammad Ashad Kabir Anik Das Sumaiya Amin Sheikh Mohammed Shariful Islam Sabiha Samad, Fahmida Ahmed. Smartphone apps for tracking food consumption and recommendations: Evaluating artificial intelligence-based functionalities, features and quality of current apps. *Science Direct*, 2022.
- [Sto24] Google Play Store. Snapcalorie ai calorie counter. *Google Play*, 2024.

- [ZMKK⁺21] Mila Zečević, Dejan Mijatović, Mateja Kos Koklič, Vesna Žabkar, and Petar Gidaković. User perspectives of diet-tracking apps: Reviews content analysis and topic modeling. *J Med Internet Res*, 2021.