



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

SZAKDOLGOZAT

**OE-NIK
2022**

Hallgató neve:
Hallgató törzskönyvi száma:

**Fülöp Viktor Róbert
T/003843/FI12904/N**

Óbudai Egyetem
Neumann János Informatikai Kar
Kiberfizikai Rendszerek Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve:
Törzskönyvi száma:
Neptun kódja:

Fülöp Viktor Róbert
T/003843/FI12904/N

A dolgozat címe:

Étrend tervező rendszer
Diet planning system

Intézményi konzulens:
Külső konzulens:

Simon-Nagy Gabriella

Beadási határidő:

2021. december 15.

A záróvizsga tárgyai:

Számítógép architektúrák
Big Data és üzleti intelligencia
specializáció

A feladat

Tervezzon és valósítson meg egy olyan rendszert, amely képes a felhasználó adatai és céljai szerint neki megfelelő étrendet összeállítani. Kutassa fel és vizsgálja meg a feladat megoldásához használható technológiákat és adatforrásokat. Ismertesse a hasonló célú rendszerek működését és képességeit. Valósítson meg egy olyan tesztrendszert, amely lehetővé teszi személyes adatok (például a kor, magasság, súly) és egyéni célok (például fogyás, ételérzékenység miatti diéta, egészségesebb táplálkozás) megadását. Legyen megadható az egyes hozzávalókra vonatkozó preferencia is az egyéni ízlés szerint. Mindezen adatok és korszerű dietetikai ismeretek alapján ajánljon az alkalmazás teljes étrendet, és tegye elérhetővé a javasolt ételek receptjeit is.

A dolgozatnak tartalmaznia kell:

- a feladat pontos ismertetését,
- a hasonló rendszerek működését és képességeit,
- a feladat megoldásához alkalmazható technológiák és adatforrások tömör bemutatását,
- a választott megoldások indoklását,
- a tervezés és megvalósítás részletes leírását,
- a tesztelés tervét és eredményeit,
- az elkészített rendszer eredményeinek részletes bemutatását és értékelését,
- a továbbfejlesztési lehetőségek számba vételét.



.....
Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022. május 14.

.....
hallgató aláírása



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun Kód: Tagozat:

Fülöp Viktor Róbert..... Esti

Telefon: Levelezési cím (pl.: lakcím):

Szakdolgozat címe magyarul:

Étrend tervező rendszer

Szakdolgozat címe angolul:

Diet planning system

Intézményi konzulens: Külső konzulens:

SIMON-NAGY GABRIELLA

Alk.	Dátum	Tartalom	Aláírás
1.	2021. 02. 26.	A téma meghatározása	
2.	2021. 03. 19.	A kutatási területek áttekintése	
3.	2021. 04. 16.	Az irodalomkutatás összegzése	
4.	2021. 05. 10.	A rendszerspecifikáció véglegesítése	

A hallgató a Szakdolgozat I. tantárgy követelményét teljesítette, beszámolóra bocsátható.

Intézményi konzulens

Budapest, 2021. május 14.



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve: FÜLÖP VIKTOR RÓBERT... Neptun Kód: [REDACTED] Tagozat: ESTI.....

Telefon: [REDACTED] Levelezési cím (pl.: lakcím): [REDACTED]

Szakdolgozat címe magyarul:
Étrend tervező rendszer.....

Szakdolgozat címe angolul:
Diet planning system.....

Intézményi konzulens: SIMON-NAGY GABRIELLA Külső konzulens:

Alk.	Dátum	Tartalom	Aláírás
1.	2022. 02. 23.	Megvalósítási lehetőségek áttekintése	
2.	2022. 03. 17.	Fejlesztési fázis ellenőrzése	
3.	2022. 04. 22.	Tesztelés ellenőrzése	
4.	2022. 05. 10.	Eredmények, összegzés	

A hallgató a Szakdolgozat II. tantárgy követelményét teljesítette, védésre bocsátható.


.....
Intézményi konzulens

Budapest, 2022. május 10.

Tartalom

1	Bevezetés	2
1.1	Egészség értés	2
1.2	Az egészség értésből egészség tudatosság	3
1.3	Táplálkozás hormonális szabályozása.....	4
2	Hasonló rendszerek vizsgálata.....	6
2.1	Kiindulópont	6
2.2	NutriComp.....	6
2.3	Palatinus Fitness Klub.....	7
2.4	Way of Life	7
2.5	Összegzés	9
3	Lehetséges adatforrások.....	11
3.1	Okostányér	11
3.2	Egészségfejlesztési iroda.....	13
3.3	Élelmiszerlánc-felügyeletért felelős Államtitkárság	14
3.4	Egyéb források	14
4	Adatbázis típusok áttekintése adattárolás céljából.....	15
4.1	Adatbázisok.....	15
4.1.1	Relációs adatbázisok.....	15
4.1.2	Kulcs-érték tárolók	16
4.1.3	Oszlopalapú tároló	16
4.1.4	Dokumentumtárolók	17
4.1.5	Gráfadatbázisok	17
5	Specifikáció	18
5.1	Cél	18
5.2	Felhasználói követelmény meghatározása	18
6	Fejlesztéshez szükséges technológiák kiválasztása	22
6.1	Adatbázis.....	22
6.2	Backend.....	23
6.3	Framework	23
6.4	Frontend	24
6.5	Fejlesztő környezet.....	25
7	Rendszerterv	26
7.1	EK modell	26
7.1.1	Táblák főbb jellemzői	28

7.2	Használati eset diagram tervezet.....	30
7.3	Rendszerterv modellezése	31
7.3.1	Nem regisztrált felhasználók interakciói	32
7.3.2	Regisztrált felhasználók interakciói	32
7.4	Felhasználható képletek	33
7.4.1	Testtömeg-index [36].....	33
7.4.2	Testzsír százalék [37].....	33
7.4.3	Energiaigény [38]	33
8	Megvalósítás	34
8.1	Adatbázis.....	35
8.2	Étrendtervező rendszer szolgáltatás modul.....	38
8.2.1	A legfontosabb maven függőségek összefoglalója [41][42].....	39
8.2.2	A felépítés részletezése	39
8.2.3	Swagger	42
8.3	Étrendtervező rendszer felhasználó hitelesítését kezelő modul [43]	44
8.4	Docker [45][46].....	44
8.5	Keycloak [47].....	45
8.6	Angular applikáció	47
8.6.1	Útvonalválasztó	48
8.6.2	Hitelesítés [49][50]	48
8.6.3	Nézet	49
8.6.4	Nézet – model	49
8.6.5	Model	49
8.6.6	Szerver kommunikáció	49
8.7	Alkalmazás bemutatása	50
8.7.1	Főoldal	50
8.7.2	Felhasználói kezelőfelület.....	51
8.7.3	Étrend.....	54
8.7.4	Receptek.....	59
8.7.5	Statisztika.....	59
9	Tesztelés.....	61
9.1.1	Végétől a végéig tartó tesztelés – E2E.....	61
9.1.2	Unit teszt	62
10	Tovább fejlesztési lehetőségek	63
11	Összefoglaló.....	64
12	Summary.....	65
13	Irodalomjegyzék	66

14	Ábrajegyzék.....	71
15	Táblázatok.....	72

1 Bevezetés

Versenyszerű sportolói múltamból eredően mindig is foglalkoztatott az egészség, és az egészséges táplálkozás. Sokszor felmerült bennem az a kérdés, hogy milyen összefüggés van az egészség és a táplálkozás között. Minél mélyebben ástam bele magam a témába, hogy milyen táplálkozási forma lenne számomra a legideálisabb, legelőnyösebb, annál jobban éreztem, hogy ennek a tudománynak az elsajátítása, illetve a saját táplálkozási forma kiválasztása nehezebb, mint gondoltam. Nagyon vékony a mezsgye a között, hogy mi az, ami egészséges és mi az, ami igazán jó. Ennek az-az oka, hogy mindannyian különbözőek vagyunk. Másképp zajlanak a biokémiai folyamatok¹, más az anyagcserénk és más a hormonális állapotunk. Ebből következik, hogy az étrendet személyre szabottan kell érteni, hogy mi az, amely számunkra, egyénileg a legmegfelelőbb.

1.1 Egészség értés

Mindennapi rutinunknak megfelelően éljük az életünket, mindaddig, amíg nem jelentkezik valamilyen egészségügyi panaszunk. Ilyenkor általában elmegyünk egy vagy több orvoshoz, akik megkezdik a különböző kivizsgálásokat, s kaphatunk a problémánkra gyógyszereket. A gyógyszerek befolyásolhatják a vércukorértékünket, a vérnyomásunkat, illetve azért is felelhetnek, hogy egyáltalán jobban érezzük magunkat. Az viszont több időbe telhet, s gyakran sajnós nem jutunk el odáig, hogy megértsük azt, miért is alakult ki az elváltozás a szervezetünkben, amely a tüneteket okozza, hogy mi a panaszaink kiváltó oka és mi az, amin szükséges lenne változtatni. Ezzel olyan megoldást kapunk, amellyel csupán a tüneteket nyomhatjuk el egy ideig, s nem megyünk „vissza” egészen oda, hogy megértsük milyen változtatásokat kellene alkalmaznunk ahhoz, hogy az életminőségünk magasabb szintet érjen el. [1][2]

¹ A sejtekben zajló felépítő és lebontó folyamatok

1.2 Az egészség értésből egészség tudatosság

Fontos, hogy megértsük, hogyan reagál a szervezetünk egyes élelmiszerek fogyasztása során. Meg kell tanulnunk odafigyelni magunkra és megpróbálni kialakítani egy olyan rendszert, amely számunkra a legalkalmasabb. Mivel mindannyian különbözőek vagyunk, nem biztos, hogy például a WHO² által ajánlott ételmisszerpiramis a legtökéletesebb számunkra.

1.1. Életminőség

Néhány példa az életminőségünkre vonatkozóan a mai világban
Stressz, krónikus fáradtság, alvászavarok, depresszió, hangulatváltozások.
Szív- és érrendszeri panaszok, magas vérnyomás és koleszterinszint.
Emésztési zavarok: reflux, puffadás, ételintolerancia, hasmenés, székrekedés.
Hormonális zavarok: pajzsmirigy működési zavarok, nemi vágy csökkenés.
Pszichés problémák: önismereti zavarok, megfelelés.
Függőségek: túlzott gyógyszerfogyasztás, dohányzás, alkohol abúzusok.

Ha olyan tápanyagokat viszünk be a szervezetünkbe, amelyek anti-tápanyagként³ működnek, akkor ez később egészségügyi problémákhoz vezethet. Anti-tápanyagként ismert például a glutén⁴, az élesztő, az adalékanyagok, amelyek általában csak kis mennyiségben kerülnek a szervezetünkbe. A kiválasztó rendszerünkől függően halmozódnak mindaddig, amíg nem jelentkeznek a panaszok. Az egészségtelen táplálkozással megteremthetjük a feltételét egy úgynevezett krízisnek, majd amikor jönne egy lelki hatás - bármilyen negatív élethelyzetből adódóan -, akkor megtörténik a „robbanás”. Ebből származik egy olyan kifejezés, hogy taposóakna. Amikor elfogyasztunk egy egészségtelen ételt, tegyük fel egy cukros fánkot, akkor gyakorlatilag előkészítjük, megalapozzuk a saját betegségünket. A taposóakna elnevezés arra utal, hogy egy-egy ilyen jellegű ételmszer szervezetünkbe való bevitelével növeljük annak az esélyét, hogy ha egy olyan élethelyzetbe kerülünk, amikor elegendő - a szó átvitt értelmében - csak rálépnünk, és felrobban. Ez azt jelenti, hogy valamilyen panaszban manifesztálódik⁵. [3][4]

² Egészségügyi Világszervezet (World Health Organization)

³ Azok a tápanyagok, amelyek a legtöbb esetben emésztési és vele összefüggésbe hozható egészségügyi problémákat okoznak.

⁴ Olyan természetes fehérje, amelyek legfőképpen a gabonafélékben találhatók meg.

⁵ Egy állapotot kimutat

Amit elvárunk az élelmiszerektől:

- Biztosítson elegendő energiát (vitalitás, agyműködés).
- Legyen tökéletes üzemanyag a sejteink számára.
- Megfelelő tápanyagot biztosítson a sejteknek. Kiemelendő, hogy minőségtelen ételeket hiába eszünk, a sejteink az értékes vitaminokat kívánják.
- Legyen mentes a káros anyagoktól.
- Legyen élvezetes és finom.

1.3 Táplálkozás hormonális szabályozása

Két hormon által történik az éhség szabályozása a szervezetünkben a ghrelin és az inzulin. Az éhséget kiváltó hormon a gyomorban keletkezik, ez a ghrelin. A ghrelin párja a leptin, amely a jóllakottság érzetet kiváltó hormon és a zsírsejtekben keletkezik. Mivel ez a hormon a zsírszövetekben termelődik, több idő szükséges arra, hogy jelezzék, ha jól laktunk.

Amennyiben az inzulin felszabadul, például amikor nassolunk, a szénhidrátok bekerülnek a sejtekbe, a maradék inzulin viszont ott marad a vérpályánkban. Ez a folyamat arra készítet minket, hogy együnk. Az inzulin párja a glukagon. A glukagon a májban tárolt glikogént tudja felszabadítani, amely a tartalékhoz fog hozzányúlni. Amikor a glikogén uralkodik, akkor az inzulin szintje lent van. Amikor fent van az inzulin szintje, akkor a glikogén van lent. Amikor az inzulin magas szinten uralkodik, akkor a szervezetünkben építkezés történik, amely azt jelenti, hogy az inzulintól hízzunk. Ebből következik, hogy az inzulinszintet alacsonyan kell tartani.

Az inzulin egy nagyon érzékeny hormon, amikor egy kis mennyiségű cukrot, vagy más rossz szénhidrátot viszünk a szervezetünkbe, akkor azonnal inzulin termelődik. Ha például csak egy szem cukrot visz be az ember a szervezetébe, akkor a szervezet nem tudja, mennyi inzulint termeljen, így ott marad a véráramban és arra programozza az embert, hogy egyen még szénhidrátot, s ilyenkor jön el az az érzés, hogy valamit még enni kell. Ezért lenne fontos, hogy étkezések között egyáltalán ne vigyünk a szervezetünkbe sem cukrot, sem szénhidrátot, mert az inzulin azonnal elkezd termelődni. A vércukorszint ingadozása magas inzulinszintet eredményez, amelynek következtében a szervezet raktározni kezd. Amióta cukorgyártás létezik, azóta túlédésítjük az ételeinket, ezáltal növekedik az elhízott emberek száma is.

Fontos megemlíteni az inzulinrezisztencia jelenségét. Ez azt jelenti, hogy a szervezetünkben állandóan magas a vércukorszint, emiatt a sejtek azt üzenik, hogy tele vannak és lezárja a receptorait, s csak néhányat hagy meg, hogy kevés tápanyag jusson be a sejtbe. Ha nem kap cukrot a szervezet, akkor elkezdődik a fogyás, elkezd nőni az inzulin érzékenység, egyre több tápanyaghoz jut a szervezet és csökken a szervezetben

például a gyulladássos hajlam. Ebből következik, hogy hozzáadott cukorra egyáltalán nincs szükségünk.

Mind az, amelyet fentebb ismertettem hosszú időbe telt, mire megértettem. Fontos kiemelnem, hogy személy szerint, mint versenysportoló mindenképpen fontosnak tartom a táplálkozás mellett a rendszeres mozgást, ellenben a szakdolgozatom témája erre nem terjed ki.

Az út, amelyet bejártam rávilágított arra, hogy milyen jó lenne egy olyan weboldalt létrehozni, amely nem csupán abban segítene, hogy egészségesebben táplálkozzak, hanem változatosan is. Ugyanis sok fejtörést okozott már az, hogy hétvégenként azon kellett gondolkodnom, mit is fogok enni a következő héten, vagy aznap. Hamar rájöttem, hogy túl egyoldalúan táplálkozom, mert mindig ugyanazt a pár receptet készítem el. Tudom, hogy ez nem egyedi probléma, mert amikor még kézbesítőként dolgoztam, több családdal történő beszélgetésem során előbukkant, hogy túl egyoldalúan táplálkoznak a családok, ellenben nincs is idejük ezen változtatni, sőt többször inkább rendelnek egy vagy két megszokott étteremből.

A szakdolgozatom során felépítetek egy weboldalt, amely változatos ételeket kínál az oda látogatóknak. A regisztrált tagoknak lehetőségük lesz különböző célokat beállítani, például fogyókúrázni szeretnének, vagy éppen az egészségmegőrzés a céljuk. Továbbá egészségügyi beállításokon keresztül, különböző egyéni érték is beállítható lesz, ebben az esetben a program, megfelelő módon helyettesíti majd a helyettesítendő élelmiszereket. Különböző riportokon keresztül a felhasználó figyelemmel kísérheti étkezési szokásait visszamenőleg. Kiemelten fontosnak tartom az étrendek generálásának egyszerűségét, amely néhány alapbeállítást követően gyorsan megtörténik. [5][6]

2 Hasonló rendszerek vizsgálata

2.1 Kiindulópont

Alapötletem az, hogy amikor végiggondolnánk azt, hogy az adott napon vagy héten mit együnk, akkor ezt helyettünk egy szoftver tegye, valamilyen módszertan szerint. Elsődleges irány az, hogy lépéseket tegyenek az emberek saját egészségük megtartása érdekében úgy, hogy közben lehetőségük nyíljon új étel vagy ételtípusok megismerésére. Ellenben a digitalizáció új korszakában az a társadalmi réteg, amely könnyen hozzáfér a digitális eszközökhöz, egyre több rendszernek a része, így egyre több rendszerbe kell belépniük, s azt esetleg karbantartaniuk, információt kinyerniük és természetesen használniuk. Ebből következik, hogy a szoftvereket egyre egyszerűbben kezelhetővé, átláthatóbbá kell tenni, annak érdekében, hogy a rendszert használó felhasználók rendszeres látogatói legyenek a szoftvernek. A témám átgondolása során megkezdtem azoknak a rendszereknek felkutatását, amelyek online étrendtervezést kínálnak.

2.2 NutriComp

Kutatásom során találtam meg a NutriComp weboldalát, amely automatikus étrendtervező szoftvereket kínál. A célcsoportok közétkeztetéssel foglalkozó állami szervek, sporttáplálkozási szakemberek, dietetikusok⁶. Tehát az általuk kínált megoldásokkal elsősorban szolgáltatókat céloznak meg. A szoftver segítségével tárolják az ügyfél adatait és anamnézisét⁷. Továbbá tárolják az ügyfelek szabadidős tevékenységét, munkavégzés típusát, akár 24 órás bontásban. Sportolók esetében az adott sportág és a hozzá tartozó intenzitás is kiválasztható. Az adatok alapján kiszámítható az alapanyagcseréje, energiája és a tápanyagszükséglete. Az ételkészítés szükséglet pedig a legújabb hazai ajánlás szerint, az okostányér alapján tekinthető meg.

További előkészítések során megadható a tervezendő étrend típusa (például fogyókúra, diabétesz, egészségmegőrzés, vegetáriánus). Kiválasztható a szezon hónapokra lebontva, és az élelmiszerek árkategóriája. Korlátozó beállításokat is végrehajthatunk a zsír, cukor, rost, illetve nátriumra kiterjedően. Kerülendő élelmiszereket rögzíthetünk annak érdekében, hogy a rendszer más élelmiszerekkel helyettesítse azokat. Végül választhatunk különböző étkezés típusokat (például hideg, meleg, kihagyható kisebb étkezések).

⁶ Olyan szakember, aki az egészséges és a beteg emberek számára nyújt segítséget a megfelelő táplálkozás kiválasztásában

⁷ Kórelőzmény

A látottak alapján a szoftver előnye, hogy rendkívül kifinomult, pontos és egyénre szabott étrendek állíthatóak elő segítségével. A célcsoportnak megfelelő lehet, ellenben hétköznapi használatra drága és túl bonyolult. Szakember nélkül nehezen kezelhető és értelmezhető, ugyanis a szoftver bárki számára elérhető, így akár egy átlagos család vagy egyén is használhatja. A szoftver képes egy grafikonon ábrázolni, hogy a tápanyagok, ásványi anyagok, energiai megoszlások, és vitaminok mennyire térnek el a szükségéstől. A grafikonon való ábrázolás mindenképpen előnyös, ellenben a bemutatóban nem tesz említést a múltbéli táplálkozási szokások kimutatására, amelynek segítségével a jelenre és a jövőre nézve is adhatna távpontot arra vonatkozóan, hogy például milyen egészségi állapot várható. Szintén előny, hogy a generált étrend nyomtatható, pdf formátumban letölthető és elküldhető e-mailben a pontos recepttel együtt.

Arra a következtetésre jutottam, hogy a NutriComp étrendtervező rendszerei egy szűk piaci rést fednek le, ellenben számukra talán a talált legjobbjaként említeném meg. [7]

2.3 Palatinus Fitness Klub

A legtöbb weboldal olyan lehetőségeket kínál, amely jellemzően egy részletes kérdőív kitöltésével kezdődik és ezen keresztül történik az étrendtervezés megrendelése, amely után személyre szabott online étrendtervezést kapunk. Ilyen például a Palatinus Fitness Klub által kínált lehetőség is. Előnye többnyire az, hogy nem szükséges személyesen is részt venni és felvenni az adatokat, hanem ezt online is megtehetjük. Ellenben ahogyan a jelenlegi weboldal is mutatja, túl sok mindent nem árul el Önmagáról a szolgáltató. Hátránya az, hogy nehézkes az azonnali újratervezés, a teljes irányítás a szolgáltatónál van. Nem kapunk részletes eredményeket vagy grafikonokat. Szintén fizetős rendszerről beszélünk. [8]

2.4 Way of Life

Kutatásom során az egyetlen ingyenesen elérhető online felület, amelyet találtam. Ahhoz, hogy az online felületet használni tudjuk, csupán regisztrálni kell. Az alkalmazás nem túl széleskörű. Első lépésként meg kell adni az étkezések számát, majd kiválasztjuk az alapanyagokat. Rögzíthetünk saját magunk is élelmiszereket, illetve ki is választhatunk olyanokat, amelyek az adatbázisban szerepelnek. Az alapanyagok mellett táblázatba foglalva megjelenik egy – egy mező, amelyben beállítható a kívánt mennyiség. Továbbá ugyanitt szerepelnek a fehérje, szénhidrát, zsír és energia mezők, amelyek az adatbázisból kiválasztott élelmiszerek esetében nem szerkeszthetők. Miután minden kívánt élelmiszert felvettünk az étkezéseinkkel kapcsolatban, tovább lépve megjelenik táblázatba foglalva az étrend, étkezésekre bontva. Receptek nem találhatóak az oldalon. Tulajdonképpen ugyanazt látjuk, mint a korábbi lépésben. Négy darab művelet érhető el,

az étrend szerkesztése, nyomtatása, letöltése és törlése. Az alkalmazás leginkább arra való, hogy aki tudni szeretné mennyi kalóriát, zsírt, fehérjét és szénhidrátot juttat a szervezetébe, azt itt megtudhatja. Előnye az, hogy a felhasználónak nem kell kutatnia az élelmiszerek tápanyagértékei után, illetve ingyenesen elérhető és egyszerű a használata. Hátrányként mindaz felsorolható, amelyeket a fenti két esetben is tapasztalhatunk, a NutriComp és a Palatinus Fitness Klub oldala esetében. [9]

2.5 Összegzés

A szakdolgozatom előző három pontjában röviden bemutatásra került három különböző típusú rendszer, amelyeket a következőkben tovább elemzek előnyeik és hátrányaik szempontjából. Ezek a pontok nagyon fontosak számomra, és a saját rendszerem kialakítása tekintetében, hogy mi az, ami jól működik a másik három rendszer esetében, és mi az, ami erősítést kíván annak érdekében, hogy egy sikeres programot hozzak létre.

2.1. Nutricomp összefoglaló

NutriComp	
Előnyök	Hátrányok
Ügyfelek adatainak pontos rögzítése (például: súly, magasság, BMI ⁸ , stb.)	Drága
Beállítható a napi testmozgás mértéke	Célcsoportja nem az átlag felhasználó
Sporttevékenységek rögzítése	Használata időigényes és bonyolult
A rögzített adatok alapján elérhetővé válik az alapanyagcsere, energia, tápanyagszükséglet	Szakemberek nélkül nehezen értelmezhető
Megadható a cél (például: fogyókúra, diabétesz stb.)	Grafikonon való ábrázolás esetében a múltbéli adatok hiányossága a táplálkozási szokások tekintetében
Az ételtípusok kiválaszthatóak árkategória szerint	
Elérhető korlátozó beállítás a zsírra, cukorra, rostra és nátriumra vonatkozóan	
Beállíthatóak kerülendő élelmiszerek	
Étkezés típusának beállítása (például: hideg, meleg stb.)	
Grafikonon való ábrázolások (például: tápanyagok, ásványi anyagok, energia megoszlások stb.)	
A generált étrendek kinyomtathatóak, e-mailben elküldhetők, pdf formátumban letölthetők	

⁸ Testtömeg-index

2.2. Palatinus Fitness klub összefoglaló

Palatinus Fitness Klub	
Előnyök	Hátrányok
Nincs személyes kontakt, online intézhető	Fizetős
A felhasználónak csupán egy kérdőívet kell kitöltenie	Az online étrend módosítása nehézkes, mivel azt egy valós személy állítja össze a kérdőív alapján
A felhasználónak nincs további adminisztratív feladata	A felhasználó nem kap részletes eredményeket

2.3. Way of Life összefoglaló

Way of Life	
Előnyök	Hátrányok
Ingyenes	A rendszer nem tartalmaz recepteket
Használata egyszerű	Csak három fő tápanyag típus látható: zsír, szénhidrát, fehérje
Átlátható	Csupán az aznapi fogyasztáshoz mérten a kalóriaszámlást könnyíti meg
Adatbázisból kiválaszthatóak az élelmiszerek	Nincs kiértékelés, például grafikon formájában
Felhasználó is rögzíthet olyan élelmiszereket, amelyek nem szerepelnek az adatbázisban	

3 Lehetséges adatforrások

3.1 Okostányér

Az egyik lehetséges adatforrásomnak bizonyult a Magyar Dietetikusok Országos Szövetsége által elérhető okostányér weboldal, ahol megbízható információkhoz jutottam. Az MDOSZ tagjai főiskolát végzett dietetikusok és egyetemet végzett táplálkozástudományi szakemberek, akik jelentős létszámban az élelmiszeripari cégek munkáját is segítik. Az MDOSZ fő célja, hogy a fogyasztók hiteles tájékoztatást kapjanak az egészséges életmódnak megfelelő táplálkozási gyakorlathoz és szokások kialakításához. Ebből következik, hogy a szervezet a lakosságot célozza meg annak érdekében, hogy segítsen számukra megőrizni egészségi állapotukat vagy javítsa azt. Továbbá segítséget nyújt a táplálkozási szokások kialakításában.

Az oldal által olyan fontos és hiteles adatokhoz jutottam, mint a testtömeg index vagy más néven Body Mass Index, az alapanyagcsere kiszámításának módja, és nem utolsósorban a fizikai aktivitás korrekciós faktorai. Továbbá ajánlott ételadagok ismertetésében is segítségemre volt, és a különböző napszakoknak megfelelően, néhány alap recepthez is hozzájutottam, amelyek ingyenesen elérhetőek. Ugyan az említett adatok kevésbé járulnak ahhoz hozzá, hogy adatként tömegesen megjelenjenek az adatbázisomban, ugyanakkor a szoftver működéséhez és alapelveihez mindenképpen hozzájárulnak. [10][32][33]

A weboldalon ingyenes letölthető rövid tájékoztatók:



3.1. ábra: Mi legyen egy nap a tányérodon



3.2. ábra: Ajánlott ételtípusok

3.2 Egészségfejlesztési iroda

Az Egészségfejlesztési iroda weboldalán ingyenesen elérhető és letölthető adatokra leltem. A Sarkadi Kistérségi Járóbeteg Szakellátó Központ Egészségfejlesztési Iroda által fenntartott oldalon a célja megegyezik az MDOSZ céljaival, amely szintén az egészséges életmódot támogató minták elterjesztése.

Az oldalon a következő adatforrások találhatók meg:

- Élelmianyagaink kalória és vitamintáblázat
- Élelmiszerek kalória táblázata
- Kész ételek tápanyagtáblázata
- Tápanyag táblázat

A letöltött Élelmianyagaink kalória- és vitamin táblázataiban közel 1600 élelmiszer és készétel található meg. Minden esetben részletezésre került számokban kifejtve a bennük található tápanyagok és a vitaminok mennyisége.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Élelmianyagaink kalória és vitamintáblázata (100g tisztított élelmiszerben)																		
2																			
3	Élelmiszer megnevezése				Kalória (kJ)	Fehérje	Zsír	Szénhidrát	Víz	Hamu	Rost	Vitaminok							
4						gramm						Karotin	A	B1	B2	Nikotinsav	C	D	
5	mgmikrogrammmgmgmikrogr																		
6	GABONÁK ÉS GABONA-																		
7	TERMÉKEK																		
8	Gabonamagvak, őrlémények																		
9	és fésztek																		
10	Árpagyöngy (gersli)				349 (1460)	9	1,4	75	11,9	1,2	1,5	0	0	250	100	2,5	0	0	
11	Búza, hántolt (buris)				343 (1435)	12,5	1	71,1	12,5	1,5	1,4	0	0	275	50	4	0	0	
12	Búzadara				340 (1423)	9,4	1	73,4	14,4	0,6	1,2	0	0	150	90	2	0	0	

3.3. ábra: Élelmianyagaink kalória- és vitamin táblázata

Ami elsőre látszott, hogy az adatforrásokat transzformálni⁹ kell, mivel minden táblázat más struktúrában jeleníti meg az adatokat. A transzformáláshoz több elérhető technológiára támaszkodhatok, így nem jelenthet problémát a különböző adatstruktúra. Ellenben mivel ingyenesen elérhető nyílt adatokról beszélünk, amelyek hivatalos forrásból származnak, felhasználhatóak a saját adatbázisomban, s ennek során tudom tesztelni a programot. [11]

⁹ Átalakítás

3.3 Élelmiszerlánc-felügyeletért felelős Államtitkárság

Az Élelmiszerlánc-felügyeletért felelős Államtitkárság weboldalát az Agrárminisztérium tartja fenn, ahol szintén ingyenes és letölthető adatforráshoz jutottam. A Földművelésügyi Minisztérium Élelmiszer-feldolgozási Főosztály, NÉBIH, OGYÉI, NAK, Campden BRI Magyarország, Magyar Hűtő- és Konzervipari Szövetség elindított egy kezdeményezést, amelynek keretében összeállításra került a gyorsfagyasztott termékek és a konzerv termékek tápanyag adatbázisa is, amely egy – egy táblázat. Ezen felül elérhető egy általános összefoglaló az élelmiszerek tápértékjelöléséről, amely részletezi a különböző elérhetőségeket, mind Magyarországi, mind Nemzetközi adatbázisok tekintetében, ugyanakkor a dokumentum 2016-os megjelenésű, és az abban szereplő hivatkozások nem mindegyike érhető el. Az elérhető magyar nyelvű online tápanyagtáblázatban átfedést tapasztaltam az Egészségfejlesztési iroda által nyújtott adatbázisban szereplő adatokkal, ellenben tartalmaz új adatokat is, amelyeket fel tudok majd használni. A felsorolt adatbázisok között szerepelnek könyvek is, ellenben nem ingyenesek, így azokat nem tudtam megvizsgálni, másrésről a benne szereplő adatok felhasználása manuális adatrögzítés folyamán történhetne meg, így ilyen irányú kereséseket nem végeztem. Továbbá a nemzetközi adatbázisok közül a Holland RIVM weboldalán [13] található egy ingyenesen letölthető táblázat, a Nevo-online request dataset, amely holland nyelven érhető el. [12][34]

3.4 Egyéb források

Az élelmiszerek tápanyag adatbázis felkutatása során a legtöbb esetben azt tapasztaltam, hogy az Élelmiszerlánc-felügyeletért felelős Államtitkárság weboldalára irányít át, így a lehetőségeim korlátozottak. Mint például a Nemzeti Agrárgazdasági Kamara, illetve Campden BRI Magyarország esetében is. [14][15]

4 Adatbázis típusok áttekintése adattárolás céljából

4.1 Adatbázisok

Az adatbázis kiválasztása nem mindig egyértelmű és könnyű feladat. Erre több adatbázis-áttekintés létezik, ellenben nem mindig adnak egyértelmű választ arra, hogy az adott alkalmazáshoz melyik típusú adatbázist válasszuk ki. Nem biztos, hogy elegendő csupán összehasonlítani az SQL és NoSQL adatbázis-kezelő rendszereket és ez alapján dönteni, hogy melyik támogatná jobban a projektünket.

4.1.1 Relációs adatbázisok

A relációs adatbázisok táblákban tárolják az adatokat. A relációnak legnagyobb eleme a tábla. Táblán belül a különböző oszlopok a tulajdonságokat jelölik, míg a sorok a rekordokat, másképpen egyedpéldányokat. Egy sornak az egyetlen eleme pedig a mező, ahol az oszlop típusának megfelelő adat helyezkedik el. A leggyakrabban használt típusok között szerepel a karakter, szám, dátum/idő. A különböző táblákban lévő sorok között valamilyen reláció, kapcsolat hozható létre. A kapcsolatok létrehozásának módja legtöbbször külső kulcsos megoldást jelent, tehát a táblákat és a bennük lévő adatokat kapcsolatokkal kötjük össze. A relációs adatbázisokban a legrugalmasabban tudom elérni az adatokat. [16][35]

Erősségek:

A relációs adatbázisok kiválóan kezelik a strukturált adatokat. Az adatok könnyen tárolhatóak és lekérdezhetőek. A tábla tartalma gyorsan növelhető és egyszerű, mert az adatok hozzáadása a meglévő adatok módosítása nélkül történik. Biztosított a konkurens használat ez azt jelenti, hogy a felhasználók azonos időben elérhetik, módosíthatják az adatokat. A felhasználók számára különböző korlátok is beállíthatóak, mint például, amikor egy adott adatbázis elérésekor a felhasználó csak olvasási jogot kap. Emiatt a relációs adatbázisok alkalmasak a többszintű hozzáféréshez.

Hátrányok:

A strukturálatlan adatok kezelése nehézkesen történik. A relációs adatbázisban létrehozott rendszerek létrehozása és karbantartása költséges. Körülményes a táblák sorainak egymáshoz történő hasonlítása, ilyen például az időben változó adatok kezelése. A különböző táblákban lévő, de összetartozó adatokat minden felhasználás esetében össze kell kapcsolni.

Relációs adatbázis-kezelő rendszerek:

Relációs adatbázisok, mint például az Oracle használata, olyan esetekben ajánlott, ahol az adatok integritása fontos, az adatok strukturáltan állnak rendelkezésre. Ilyen területek lehetnek például a könyvelési, pénzügyi és banki, a leltárkezelési, tranzakciókezelési rendszerek esetében.

4.1.2 Kulcs-érték tárolók

A kulcs-érték típusú adatbázisok kulcs-érték párokban tárolják az adatokat. Adatszerkezetük asszociatív tömbökre nézve, szótárakra hasonlít a leginkább. A kulcs egy egyedi azonosító, amely csakis az értékhez kapcsolódik. A tárolt érték lehet szám, karakterlánc, számláló, JSON, XML, HTML, PHP, bináris, kép, rövid videó, lista stb. Strukturálatlan adatok tárolására is alkalmas. Erőssége, hogy rugalmas, könnyen kezelhető. A kulcsok össze vannak kapcsolva az értékekkel, így gyors elérésű, s ebből következik, hogy ha nincs index keresés a teljesítmény magas. A kulcsok rendezése, illetve a tartományok kezelése hatékony. Horizontálisan skálázhatóság miatt sok, olcsó gépből nagy teljesítmény érhető el. Hátránya, hogy kevés művelet támogatott. Csak akkor gyors, ha ismerjük a kulcsot, illetve a lekérdezés is csak ez alapján működik. Ilyen megoldást alkalmaz például az Instagram és a Twitter is. Kulcs-érték tároló például a Redis. [17]

4.1.3 Oszlopalapú tároló

Oszlopalapú tároló adatbáziskezelő rendszer, mint például a HBase úgy néz ki, mint egy elforgatott hagyományos adatbázis. Több gépen szétszított adatmennyiséget tárol és dolgoz fel. Dinamikus, első sorban komplex, és változó tartalmú rekordok relációs tárolására ideális. Az oszlopalapú adatbázis-kezelők is relációkat tárolnak, azonban ezek a relációk nem feltétlen normál formában tárolódnak, illetve a redundancia is lehetséges. Csupán a fizikai szervezésben a sor- és oszlopalapú szervezés tér el. Előnye, hogy horizontálisan skálázható, így sok olcsó gépből nagy teljesítmény érhető el. Az oszlopos adatbázisok jobban tömörülnek, mivel null értékű mezőket nem tárolnak, így gazdaságosabbak, mint a rekord alapú rendszerek. A nem strukturált adatokat is jól kezeli. Hátránya, hogy kevés művelet támogatott. Csak akkor gyors, ha ismerjük a kulcsot és a lekérdezés is csak index alapján működik. Hiányoznak belőle a relációs adatbázisokra (RDBMS) jellemző tulajdonságok, mint például a típusos oszlopok, másodlagos indexek, triggerek. Tehát nem tekinthető hagyományos értelemben vett adatbázisnak. [18]

4.1.4 Dokumentumtárolók

Dokumentumtároló, mint például a MongoDB a JSON dokumentumokat bináris kódolású formátumban, BSON formátumban kezeli. A BSON egy számítógépes adatsere-formátum a JSON bináris formátuma. Többféle típusú adatot képes tárolni ilyenek például a számok, dátumok, tömbök, objektumok, sztringek. [19]

4.1.5 Gráfadatbázisok

Gráf alapú adatbázisok alkalmazását tipikusan a tömegközlekedési hálózatok térképei, navigáció vagy családfa esetében használják. Abban az esetben érdemes használni, amennyiben az adatok közötti kapcsolat áll a középpontban. A gráfbejárás során alapvetően elkezdünk egy csúcstól felkutatni a szomszédos csúcsokat. Két csúcs akkor nevezhető szomszédosnak, amennyiben vezet közöttük él. Erre kettő alapvető stratégia létezik a szélességi és mélységi keresés. A szélességi keresés során megnézzük, hogy az egyes csúcs milyen csúcsokat ismer, majd visszalépünk a kettes csúcsig és ezt követően megvizsgáljuk azt, hogy „Ö” milyen csúcsokat ismer és így haladunk tovább. A mélységi bejárás ezzel szemben, amikor az egyes csúcsból kiindulva megkeressük a célt. Végig megyünk a legmélyebb pontig. Minden olyan esetben, amikor egy zsákutcahoz érünk visszalépünk és újra a legmélyebb pontig haladunk mindaddig tesszük ezeket a lépéseket, amíg meg nem találjuk a célt. Gráfadatbázis kezelő rendszer például a Neo4J. [20]

5 Specifikáció

5.1 Cél

A szoftver célja egy olyan webalkalmazás, amely lehetőséget nyújt a felhasználóknak étlapot tervezni, mindezt úgy, hogy megismerjenek új és egészséges ételeket. A regisztrált tagoknak lehetőségük nyílik különböző célok beállítására, és ezen keresztül beállítható például az egészség megőrzése, vagy a fogyás. Különböző elemzéseken keresztül a felhasználók figyelemmel kísérhetik étkezési szokásaikat visszamenőleg is.

5.2 Felhasználói követelmény meghatározása

A nem regisztrált látogatók számára is biztosítani kell a BMI, energiaigény és testzsírszázalék kalkulátorokat. A nem regisztrált felhasználók adatai azonban nem kerülnek mentésre. Továbbá elérhetővé kell tenni az oldal látogatói számára a kisokos menüpontot. A menüpontban található majd olyan fontos információk, amelyek segítségével a felhasználó szélesítheti a helyes táplálkozással kapcsolatos tudástárát.

A regisztráció, és néhány rövid lépést követően alkalmasnak kell lennie a szoftvernek arra, hogy maximum hét napra vetítve recepteket generáljon. A regisztrálást követően a belépett felhasználók csak akkor végezhetnek étrend generálást, amennyiben a profil adatok kitöltésre kerültek. Ugyanis a profilban megadott paraméterek alapján kerül kiszámításra a BMI, energiaigény és a testzsír százalékos kiszámítása. Ennek oka, hogy az étrend generáláshoz szükséges adatból dolgozik a rendszer, amelynek az egyik alap eleme az energiaigény értéke. A BMI, energiaigény és testzsír százalékos értékeit a felhasználó belépését követően a főoldalon elérhetővé kell tenni. A születési év és nem csak egyszer, minden más paraméter naponta csak egy alkalommal adható meg annak érdekében, hogy maximalizáljuk a statisztikai adatok „helyességét”. Biztosítani kell a felhasználó számára az étrend tervezést, ahol néhány lépést követően generálhat étrendet magának. Amennyiben nem szimpatikus számára a kapott eredmény, akkor a már megadott adatok alapján újra elindíthatja a generálást. A generált étrendeket elmentheti és törölheti. A felhasználó eldöntheti, hogy a generált étrendeket milyen sorrendben kívánja a naptárjába beilleszteni.

A regisztrált tagoknak láthatóvá kell tenni a beépített receptek listáját.

A nem regisztrált tagok a következő funkciókat érhetik el:

- Bejelentkezés
- Regisztráció
- Kalkulátorok
- Kisokos

Belépést követően a következő funkcióknak kell elérhetővé válniuk:

- Profilom
 - Információk
 - Saját adatok
- Étrend
 - Étrend ütemezés
 - Étrend tervezés
 - Étrendjeim
- Receptek
- Statisztikák
 - Vitaminok
 - Makrotápanyagok
 - Ásványi anyagok
 - Saját adatok
- Kisokos

Kalkulátorok:

A regisztrált és nem regisztrált felhasználók egyaránt találkoznak majd a testtömeg-index, testzsír százalék, energiaigény kalkulátorokkal. A már regisztrált felhasználók esetében a kalkuláció a profilban megadott értékek alapján automatikusan kerülnek mindig kiszámításra. Az oldalt csupán látogató nem regisztrált felhasználóknak ellenben minden egyes esetben meg kell adniuk a paramétereit, majd ezt követően végezhetik el a kalkulációt.

Profilom:

A profilon belül az információk és saját adatok jelennek majd meg. Az információk almenüben automatikusan megjelenik majd a testtömeg-index, testzsír százalék és az energiaigény, abban az esetben, ha kitöltésre kerültek a megfelelő paraméterek. A saját adatok almenüpontban megadható adatok között szerepel például a születési idő, nem, magasság, súly, aktivitás stb. Fontos, hogy egyes adatok, például a nem és a születési idő, ne legyenek módosíthatóak.

Étrend:

Az étrend tervezés almenüpontban legyen lehetősége a bejelentkezett felhasználónak megadni olyan értékeket, amelyek befolyásolják a receptek kiválaszthatóságát az étrend generálása során. A receptek kiválaszthatóságának befolyásolásához a következőknek kell megfelelni:

- Mi a célkitűzés az adott étrend kapcsán (fogyás, súly megtartás, súly növekedés).
- Az adott receptek milyen árkategóriába esnek (olcsó, normál, drága).
- Olajokkal kapcsolatos preferencia lehetőség kiválasztása (lenmagolaj, dióolaj stb.).
- Cukorral kapcsolatos preferencia lehetőség kiválasztása (cukor nélkül, korlátozott stb.).
- Étkezéssel kapcsolatos preferencia lehetőség kiválasztása (tízórai nélkül vagy uzsonna nélkül).
- Egytől hét napig terjedő érték kiválasztása ahhoz, hogy a program hány napra tervezze meg az étrendet.

Az étrend generálását követően egy napokra lebontott étlap jelenik meg. Amennyiben vannak olyan értékek, amelyeket a felhasználó nem kíván megadni, akkor erre lehetőséget kell biztosítani. Ez alól kivételt képez a célkitűzés, az árkategória, illetve az, hogy hány napra szeretne a felhasználó tervezni. Amennyiben az adott étrend nem nyeri el a felhasználó szimpátiáját, akkor ez esetben lehetőséget kell teremteni arra, hogy újra tudja generáltatni az étrendet úgy, hogy a beállítások megmaradjanak. Ha az adott étrendet megfelelőnek találja a felhasználó, akkor lehetőséget kell adni annak elmentésére.

Fontos, hogy azok az étrendek, amelyek elmentésre kerültek, legyenek láthatók az étrendjeim almenüpontban. Az adott étrenden belül láthatóvá kell tenni annak tartalmát, tehát azt, hogy milyen receptek szerepelnek az adott napszakok szerint. Lehetőséget kell kínálni a felhasználó számára, hogy a kiválasztott étrendet az ütemezéshez hozzáadja és amennyiben, ha úgy kívánja, akkor elvégezhesse a törlést.

Az étrend ütemezés almenüpont alatt történik az étrendek naptárba helyezése. A naptárba helyezés során jelenleg elegendő, ha a felhasználó kiválasztja a számára tetsző étrendet és hozzáadja a naptárhoz. A felhasználónak lehetősége nyílik arra, hogy a napi étrendet kiválasztva tájékozódhasson a receptek pontos paramétereiről, hogy az adott recepthez milyen alapanyagok szükségesek és hogyan történik annak elkészítése.

Receptek:

Lehetőséget kell adni a felhasználónak, hogy az adatbázisban lévő recepteket megtekinthesse. A receptek alatt elérhetővé kell tenni, hogy milyen alapanyagokból és hogyan készül a recept.

Statisztikák:

A program egyik célja elemzések készítése az összeállított étrendek alapján, mint például makrotápanyagok, vitaminok vagy makrotápanyagok „bevitt” mennyisége. Az elemzést a könnyebb áttekintés érdekében, vizuális formában szükséges megjeleníteni.

Kisokos:

Itt olyan információknak kell szerepelnie, melyek az étrendtervezéssel kapcsolatban, a felhasználó egészségtudatosságát segíti elő.

A saját ötletem és a felkutatott étrendtervező alkalmazások által gyűjtött információk alapján felállítottam az alább látható a táblázatot, amelyben bemutatásra kerülnek a saját alkalmazásom jelenlegi, tervezési fázisában lévő előnyei és hátrányai.

5.1. Saját fejlesztésű online felület előnyei és hátrányai

Saját fejlesztésű online felület	
Előnyök	Hátrányok
Ingyenes	Nem áll rendelkezésre olyan szakember, mint például egy dietetikus
Már a regisztráció után az elemzéshez szükséges adatok rögzítésre kerülnek (BMI, magasság, súly)	Nem kerül rögzítésre teljes kórelőzmény
Változatos receptek, figyelembe véve a felhasználó kívánságát	Jelenleg nem áll rendelkezésre elegendő háttéradat, ilyenek például receptek és tápanyag táblázat, mivel többnyire ezek fizetős szolgáltatásokon keresztül érhetőek el
Étrendek generálása	
Korlátozó beállítás elérhetősége: zsír, cukor, rost	
Célok beállítása (például fogyókúra)	
Olyan grafikonok előállítása, amely segítségével az étkezési szokások alapján felmérhető, milyen tápanyagokhoz jut leginkább a szervezet, s melyekhez kevésbé	
Célcsoportja inkább a hétköznapi emberek	

6 Fejlesztéshez szükséges technológiák kiválasztása

6.1 Adatbázis

A feladat jellegéből adódóan, a PostgreSQL relációs adatbázis-kezelő rendszert választottam. A választásom egyik fő oka az volt, hogy vállalati szintű, ingyenesen elérhető, nyílt forráskódú fejlett rendszer. Támogatja az SQL-t, JSON-t, a relációs és nem relációs lekérdezésekhez, a bővíthetőség és az SQL megfelelés érdekében. A PostgreSQL támogatja a fejlett adattípusokat és a teljesítményoptimalizálási szolgáltatásokat, amelyek csak drága kereskedelmi adatbázisokban érhetők el, mint például az Oracle és az SQL Server.

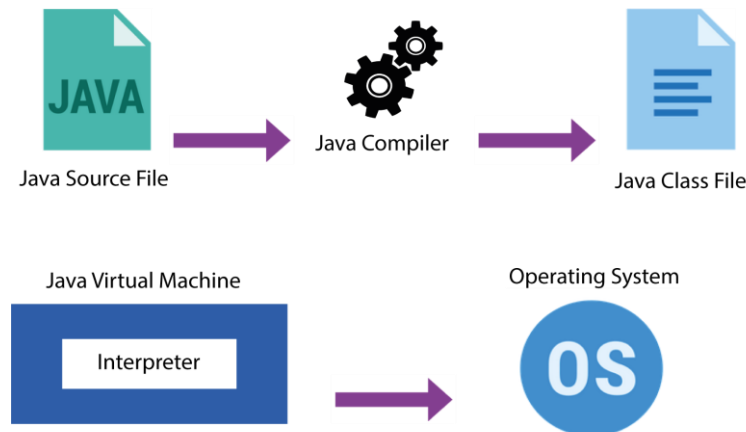
6.1. MySQL és PostgreSQL összehasonlítása

MySQL és PostgreSQL adatbázis-kezelők összehasonlítása	
MySQL	PostgreSQL
Legnépszerűbb	Legfejlettebb
Nyílt forráskódú, de most az Oracle Corporation tulajdonában van, és több fizetett kiadást kínál	Nyílt forráskódú és ingyenes szoftver
A MySQL csak akkor képes ACID-kompatibilisra, ha NDB és InnoDB Cluster Storage motorokkal használja	A PostgreSQL teljesen ACID-kompatibilis
A MySQL jól teljesít olyan OLAP és OLTP rendszerekben, ahol csak az olvasási sebesség a fontos	A PostgreSQL teljesítmény azokban a rendszerekben működik a legjobban, amelyek összetett lekérdezések végrehajtását igénylik
A MySQL megbízható és jól működik a BI (Business Intelligence) alkalmazásokkal, amelyeket nehéz elolvasni	A PostgreSQL jól működik a BI alkalmazásokkal. Ez azonban jobban megfelel adattárolási és adatelemzési alkalmazásokhoz, amelyekhez gyors olvasási és írási sebességre van szükség
Indexelés B-fa, hash, cluster indexek utóbbi esetben csak innoDB	Indexelés b-fa, hash, partial index
Támogatja a felhő szolgáltatás	Támogatja a felhő szolgáltatás

A fent felsorolt összehasonlításokon túl a PostgreSQL hátránya az, hogy a sebesség javítása érdekében végzett változtatások több munkát igényelnek, mint a MySQL, mert a kompatibilitásra összpontosít. Számos nyílt forráskódú alkalmazás támogatja a MySQL-t, ellenben előfordulhat, hogy a PostgreSQL-t nem támogatja. Teljesítménymutatók tekintetében a PostgreSQL lassabb, mint a MySQL. [21][22]

6.2 Backend

Az alkalmazás backend oldali programozás nyelvének a JAVA-t választottam, mert platformfüggetlen. A JAVA egy általános célú objektum orientált programozási nyelv. Általános célú, mert készíthetünk asztali alkalmazást, webszervert, REST API szolgáltatás vagy komplett web rendszert. A JDBC tulajdonságával egységes elérési felületet biztosít a relációs adatbázis-kezelő rendszerekhez, így a PostgreSQL-hez is. A kódot elegendő egyszer megírni és ezt követően bármely platformon futtatható, mint például Windows, Linux, Mac, amelyről a Java Virtual Machine (JVM) gondoskodik.



6.1. ábra: JVM saját készítésű ábra

Oracle honlapjáról letölthető [22] az operációs rendszernek megfelelő JAVA verzió. Ezután a JAVA keretrendszer (JVM) képes lesz értelmezni a JAVA-ban megírt programot, amely .jar kiterjesztésű futtatható állomány. Amíg az .exe kiterjesztésű fájl a windows által kerül lefuttatásra, addig a .jar a JVM által. A JVM-et pedig a windows futtatja. Így lesz platform független (platform independent) a JAVA-ban megírt program, mert többféle platformon képes a JVM. [23][24][25]

6.3 Framework

A framework kiválasztása során a Spring boot-ra esett a választásom, mert ingyenes, nyílt forráskódú rendszer. A Spring egy keretrendszer nagyvállalati alkalmazások fejlesztéséhez. Több komponenst is integrál magába, néhány fontosabb közülük:

- Inversion of Control, (IoC)
- Adatbázis elérés: egyszerűsített JDBC, JPA, Data JPA
- Üzenetkezelés
- Biztonság
- MVC webes alkalmazások

A Spring webalkalmazás nem tartalmaz mindent egyszerre, hanem a modulokat szükség esetén egyesével kell hozzáadni. Elérhetőek előre elkészült modulok is. A Spring programok futtatásához nem szükséges külső keretrendszer, beépítve találunk hozzá modulokat, amelyek önmagában futtathatóvá (self-contained) teszik.[26][27] Az egyik fő ok amiért a spring boot-ot választottam az, hogy az általa nyújtott annotációk segítségével könnyíti a fejlesztést. Ebből az következik, hogy elegendő a projekt feladat megoldására koncentrálnom.

6.4 Frontend

A weblap dizájn fejlesztését az Angular keretrendszerrel tervezem elkészíteni. Az Angular egy applikációs dizájn keretrendszer és fejlesztői platform, amelynek segítségével könnyebben és effektíven tudunk single-page (frontend) applikációkat fejleszteni.

Előnye, hogy a keretrendszer reaktív, amely röviden azt jelenti, hogy fel vagyunk iratkozva a változásokra, amelyek lehetnek például propertyk, adatbázis táblák vagy cellák. A változás során létrejön egy adatfolyam (data stream), amelyet bárhol felhasználhatunk a frontenden. A változás a GUI-n¹⁰ is rögtön változást fog eredményezni, nem szükséges azzal törődnünk, hogy mikor frissítsük az oldalt.

Legújabb verziói TypeScript nyelvre épülnek, amely egy típusos javascript superset. Éppen ezért használhatjuk a típusos kódolás minden előnyét, biztonságosabb kódot írhatunk, kevesebb a javascriptben ismert silent fail-ek esélye és fordítási időben kiszűrhető a legtöbb hiba. Több integrált könyvtára van, amelyek már eleve out of the box megoldásokat szolgáltatnak például: routing, form menedzsment/validáció, kliens-szerver kommunikáció. Saját CLI¹¹ biztosítja a kényelmes fejlesztést, build folyamatot, tesztek írását és futtatását. Szétválaszthatjuk a frontend komponensek összetevőit és teljesen függetlenül fejleszthetjük őket (HTML templatek, CSS stílusok, szkriptek). Moduláris felépítésű, írhatunk saját modulokat, illetve használhatjuk csak azokat a modulokat is, amelyekre éppen szükségünk van.

Szolgáltatások (services) egy külön layer a frontend üzleti logikához, dependency injectiont használ. Írhatunk sajátokat, de sok beépített szolgáltatás is elérhető (például HttpClient). Google által folyamatosan támogatott és naprakész.

A hátránya közé sorolható, hogy a tanulási görbe nem annyira meredek, azaz több idő szükséges a megtanulásához és megértéséhez, összehasonlítva például a Vue.js-szel, amely szintén egy reaktív javascript keretrendszer. Illetve hátránya, hogy már meglévő projektbe nehezen integrálható. [28]

¹⁰ Grafikus felhasználói felület

¹¹ Command Line Interface (parancssoros felhasználói felület)

6.5 Fejlesztő környezet

Eclipse

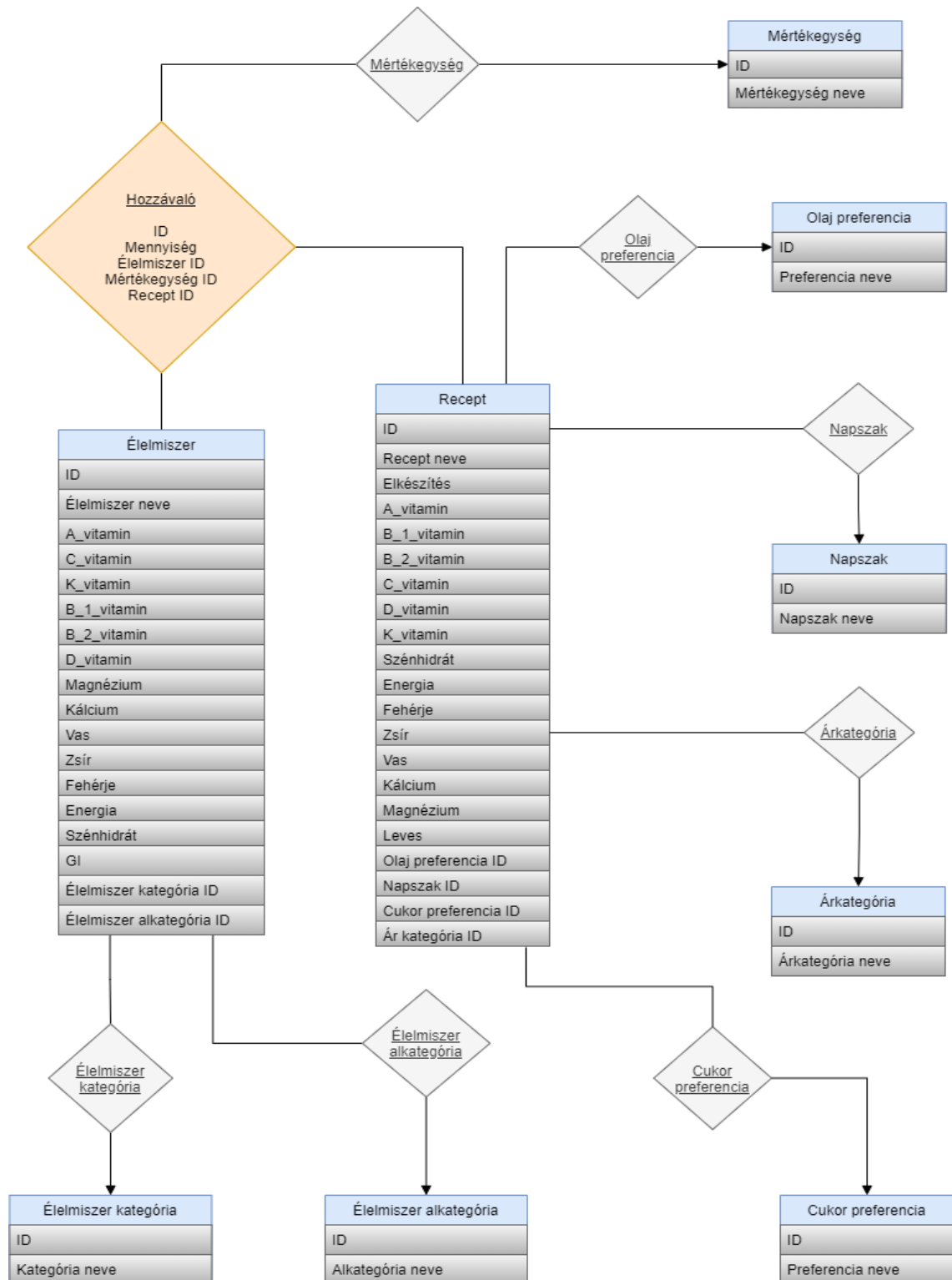
Az Eclipse egy ingyenesen elérhető Java integrált fejlesztői környezet. Több fejlesztői környezet létezik a piacon és mindegyikük hatékony a fejlesztési munka során. Az Eclipse azért jó választás számomra, mert alacsony a gépigénye és ingyenes. Az Eclipse a marketplace segítségével telepíthető a Spring Tools 4, amelynek segítségével a Spring boot alkalmazás futtatható Spring Boot App-ként.

Visual Studio Code:

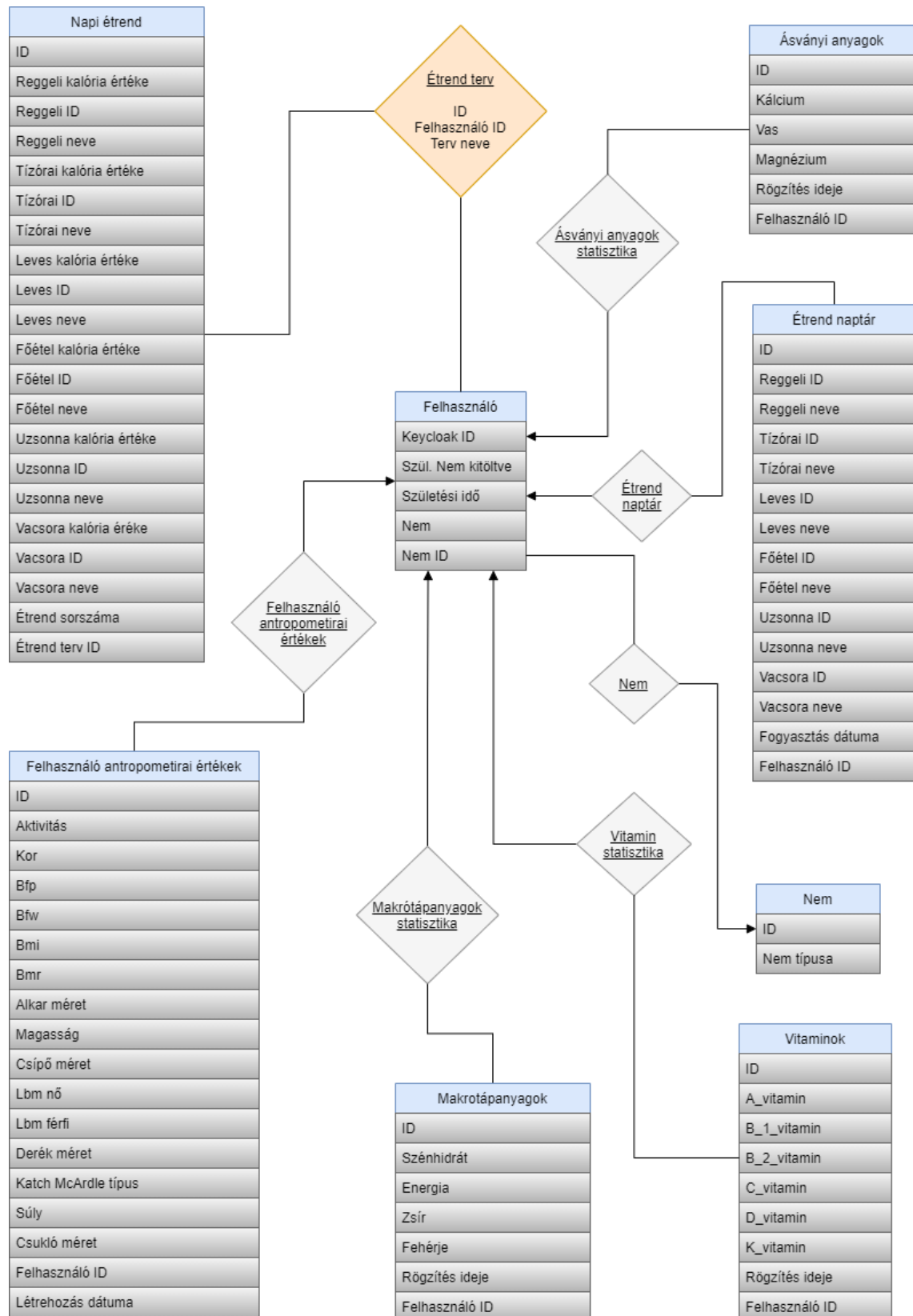
A frontend fejlesztéséhez a Visual Studio Code-ra esett a választásom, amely egy ingyenes, nyílt forráskódú kódszerkesztő, amelyet a Microsoft fejleszt Windows, Linux és OS X operációs rendszerekhez. Számos programozási nyelvet támogat és mára nagy népszerűségnek örvend a fejlesztők körében. Nagyon sok bővítmény telepíthető, mint például az Angular. [30]

7 Rendszerterv

7.1 EK modell



7.1. ábra: Étrendtervező recept EK modell tervezete



7.2. ábra: Étrendtervező felhasználó EK modell tervezet

7.1.1 Táblák főbb jellemzői

Recept EK modell:

- **RECEPT:** A recept tábla tartalmazza a vitaminok, ásványi anyagok összértékét és azt, hogy a recept milyen napszaknak, árkategóriának, olaj és cukor preferenciának felel meg. Továbbá itt található a recept elkészítésének leírása is és az, hogy milyen élelmiszereket tartalmaz.
- **OLAJ PREFERENCIA:** Olyan olaj preferencia megnevezése kerül eltárolásra, mint például az olíva vagy kókusz típusú olajok vagy zsírok.
- **CUKOR PREFERENCIA:** Olyan cukor preferencia megnevezés kerül eltárolásra mint például az eritrit, vagy cukor nélkül.
- **NAPSZAK:** Ebben a táblában az adott nap étrend szerinti napszakjai jelennek meg.
- **ÁRKATEGÓRIA:** A táblában néhány olyan érték kerül eltárolásra, amely által beazonosítható az adott étrend, hogy milyen árkategóriába esik.
- **HOZZÁVALÓ:** A tábla tartalmazza azt, hogy melyik recepthoz tartozik, milyen és mennyi élelmiszert használtunk fel a recepthoz.
- **MÉRTÉKEGYSÉG:** A hozzávalók táblában nem csak a mennyiségét, hanem a mennyiség mértékegységét is jelöljük. A mértékegység táblában a mértékegységek megnevezései kerülnek eltárolásra.
- **ÉLELMISZER:** Az adott élelmiszer ásványi és vitamin értékei jelennek meg a táblában. Továbbá itt kerül megjelölésre, hogy az adott élelmiszer milyen kategória és alkategóriába esik.
- **ÉLELMISZER KATEGÓRIA:** A táblában olyan megnevezések kerülnek eltárolásra, amely azt jelzi, hogy milyen csoportba tartozik az adott élelmiszer (például tejtermék).
- **ÉLELMISZER ALKATEGÓRIA:** A táblában olyan megnevezések kerülnek eltárolásra, amelyek azt jelzik, hogy az adott kategóriába eső élelmiszer milyen alkategóriába tartozik (például tejterméken belül tejek).

Felhasználó EK modell:

- **FELHASZNÁLÓ:** A táblában lévő ID értéke megegyezik a keycloak szervertől érkező ID értékével. Eltárolásra kerül továbbá minden olyan adat a felhasználóról, amely az étrend tervezéshez szükséges.
- **ÁSVÁNYI ANYAGOK:** A táblában eltárolásra kerülnek a naptárban lévő receptek ásványi anyag értékei.
- **ÉTREND NAPTÁR:** A felhasználó által kiválasztott étrend kerül elmentésre. Egy sor tartalmazza a receptek kulcsait és azok kalória összértékét, étkezés napi rendje szerint.
- **NEM:** A felhasználók nemei kerülnek eltárolásra.
- **VITAMINOK:** A táblában eltárolásra kerülnek a naptárban lévő receptek vitamin értékei.
- **MAKROTÁPANYAGOK:** A táblában eltárolásra kerül a naptárban lévő receptek makrotápanyagok értéke.
- **FELHASZNÁLÓ ANTROPOMETRIAI ÉRTÉKEI:** A felhasználónak mindennap van alkalma rögzíteni a paramétereit. A megadott paraméterek alapján a rendszer kiszámítja például a testtömeg indexet, a testzsír százalékot. A kiszámítást követően a megadott paraméterek és a rendszer által kiszámított értékek kerülnek elmentésre a táblában.
- **ÉTREND TERV:** A felhasználónak van lehetősége az általa kiválasztott igényeinek megfelelően generáltatni a rendszerrel étrendeket, amelyek ebben a táblában kerülnek rögzítésre.
- **NAPI ÉTREND:** Egy étrendhez több nap is tartozhat. A táblában napi bontásban, az étkezés napi rendjének megfelelő ételek egyes értékei szerepelnek.

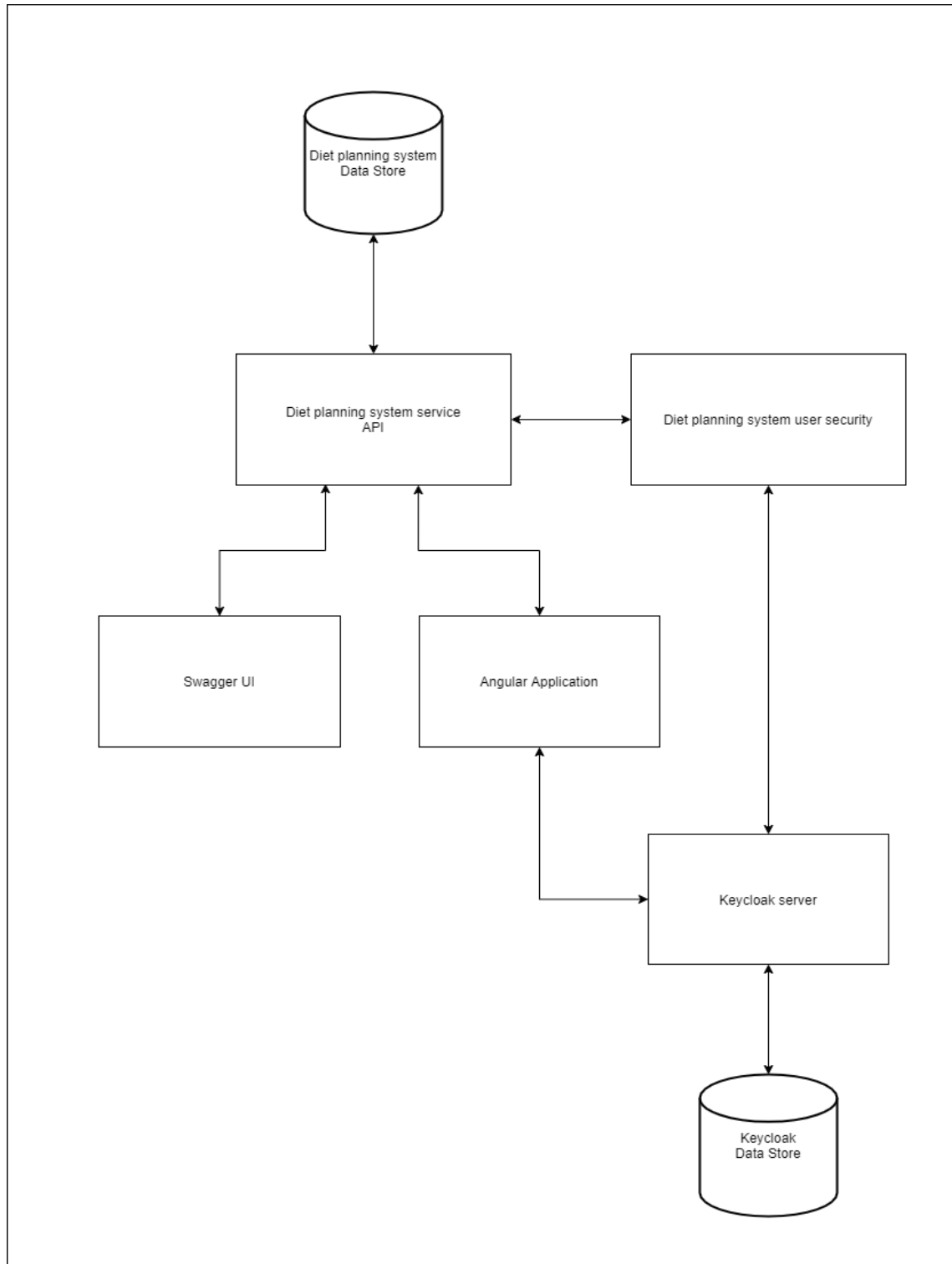
7.2 Használati eset diagram tervezet

Use Case Diagram segítségével és a specifikáció alapján bemutatásra kerül, hogy milyen interakciók várhatóak az alkalmazás megvalósítása során.



7.3. ábra: Use Case Diagram

7.3 Rendszerterv modellezése



7.4. ábra: Rendszerterv modellezése

Ahogy a használati eset diagram ábráján is látható, a rendszer egyik alkotó eleme a felhasználó bejelentkeztetése és jogosultságának ellenőrzése. A felhasználók kezelésére a Keycloak nyílt forráskódú szoftvert fogom használni, amelynek segítségével lehetővé tehetem az egyszeri bejelentkeztetést és ellenőrizhetem a felhasználó jogosultságát. A Keycloak biztosítja továbbá a regisztrációhoz és bejelentkezéshez szükséges felületet is. Rendszerterv modell alapján a következő pontokban röviden ismertetem a rendszer működését.

7.3.1 Nem regisztrált felhasználók interakciói

A nem regisztrált felhasználó megteheti, hogy adatai elmentése nélkül, ellenben annak megadásával kiszámoltassa a rendszerrel például a testtömeg indexét. Ebben az esetben a kliens oldal API-n keresztül megszólítja a szerver oldali alkalmazást, amely a megadott adatok alapján kiszámolja és visszaadja a kliensnek a várt eredményt. Jelen esetben nem történik kommunikáció az adatbázis felé, mert nem kerül eltárolásra egyetlen adat sem. A nem regisztrált felhasználók által látható minden kalkulátor a fenti alapelven működik. Kisokos és a főoldal tartalma megjelenítésért a kliens oldal felelős.

7.3.2 Regisztrált felhasználók interakciói

A regisztrált és nem regisztrált felhasználó egyaránt elsőként a Keycloak által biztosított felületet éri el. Regisztrálást követően a belépés úgy történik, hogy a kliens oldal kommunikál a Keycloak rendszerrel. Keycloak a sikeres belépést követően visszaküld a kliens oldalnak egy hitelesítő token-t, amelyet a kliens oldal minden kérés esetén átad a szerver oldalnak. A szerver oldalhoz beérkezett kérés esetén a szerver minden egyes olyan esetben kommunikál felhasználó biztonságáért felelős modullal, ahol fel lesz rá készítve. A felhasználó biztonságért felelős modul kommunikál majd a Keycloak-al, a Keycloak pedig visszaküldi az eredményt a hitelesítésről. Amennyiben sikeresen megtörténik az autentikáció¹², akkor ez esetben visszaküldésre kerül a felhasználó elsődleges kulcsa is, amelyet a saját adatbázisomban fogok majd felhasználni. Továbbá a szerver oldal kommunikál majd az adatbázissal is.

¹² Hitelesítési eljárás

7.4 Felhasználható képletek

Az weboldalt a következő számításokra szeretném felkészíteni:

- Testtömeg-index¹³ – Body Max Index (BMI)
- Testzsír százalék¹⁴ – Body Fat Percentage (BFP)
 - Testzsír tömeg – Body Fat Weight (BFW)
 - Zsírmentes tömeg - Lean Body Mass (LBM)
- Energiaigény¹⁵ - Basal Metabolic Rate (BMR)

7.4.1 Testtömeg-index [36]

$$BMI = \frac{testtömeg[kg]}{testmagasság^2[cm^2]}$$

7.4.2 Testzsír százalék [37]

$$BFP = \left(\frac{BFW}{Súly} \right) * 100$$

$$BFW = Súly - LBM$$

$$LBM (Nő) = (Súly * 1,082) + 8,987 + \left(\frac{Csukló}{3,140} \right) - (Derék * 0,157) - (Csípő * 0,249) + (Alkar * 0,434)$$

$$LBM (Férfi) = (Súly * 1,082) + 94,42 - (Derék * 0,157)$$

7.4.3 Energiaigény [38]

$$BMR (Férfi) = 66,47 + (13,75 \times \text{testsúly kg-ban megadva}) + (5,003 \times \text{magasság cm-ben megadva}) - (6,755 \times \text{életkor években megadva})$$

$$BMR (Nő) = 655,1 + (9,563 \times \text{testsúly kg-ban megadva}) + (1,85 \times \text{magasság cm-ben megadva}) - (4,676 \times \text{életkor években megadva})$$

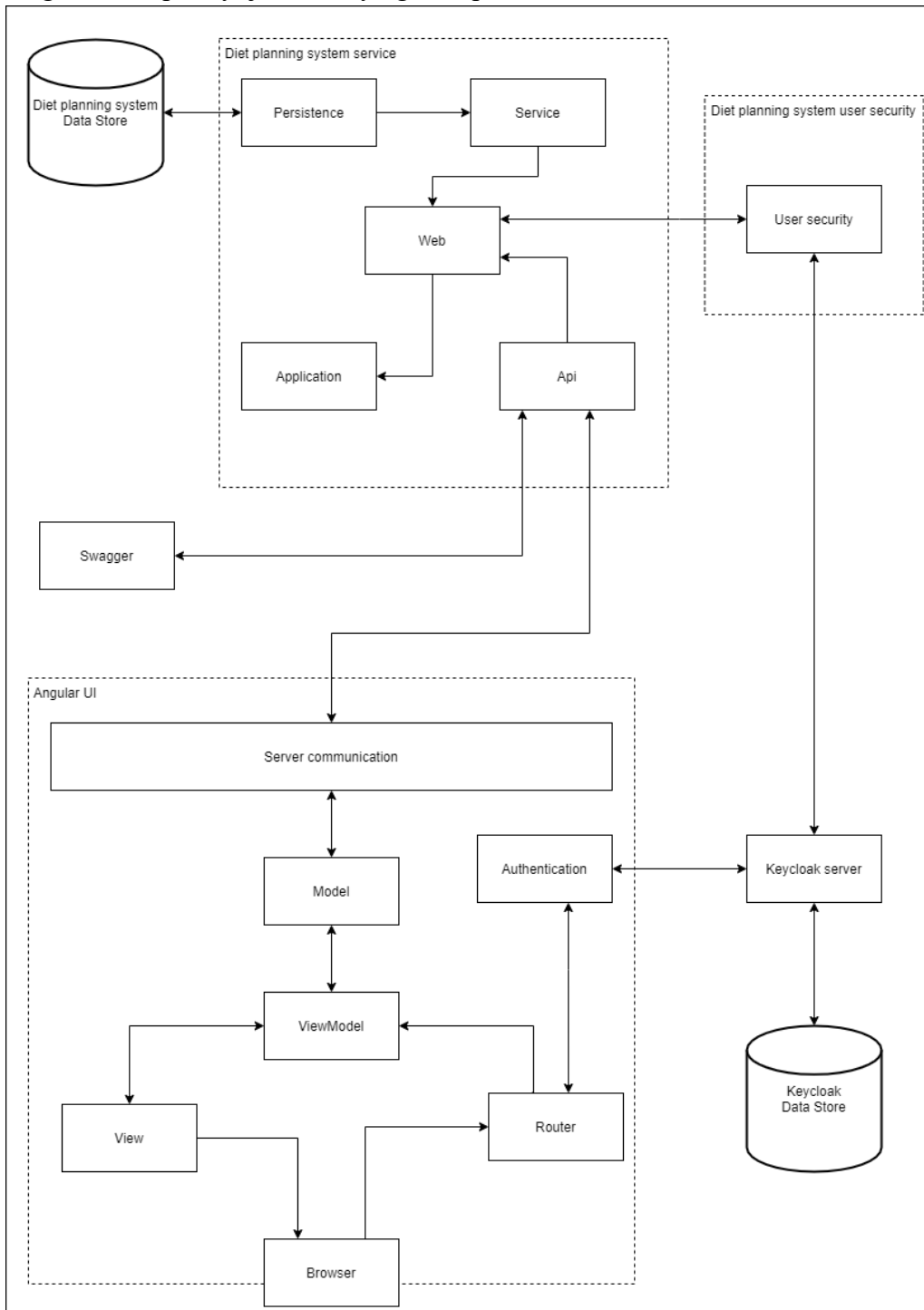
¹³ Az az érték, ami egy ember testmagasságának és testsúlyának hányadosából jó iránymutatót ad az illető egészségi állapotáról

¹⁴ A zsírszövet aránya a teljes testtömeghez viszonyítva

¹⁵ Az az érték, amely megmutatja, hogy mennyi kalóriára van szükség naponta a testsúly megőrzéséhez

8 Megvalósítás

A konkrét megvalósítás előtt elkészítettem a részletesebb moduláris tervet, amely megfelelő alapot nyújtott a tényleges implementációhoz.



8.1. ábra: Alkalmazásom moduláris felépítése

8.1 Adatbázis

Az adatbázis felépítését az étrendtervező rendszer szolgáltatás modulban építettem fel Hibernate segítségével, amelyet a következő fejezetben részletezek majd. Jelen fejezet témája a letöltött adatok generálása és transzformációja. A specifikáció alapján az adatbázis egyes tábláiból csak kiolvasás történik, azaz rendelkezünk kell megbízható adatok halmazával.

A következő táblák érintettek:

- Mértékegység
- Olaj preferencia
- Cukor preferencia
- Napszak
- Árkategória
- Élelmiszer alkategória
- Élelmiszer kategória
- Élelmiszer
- Recept

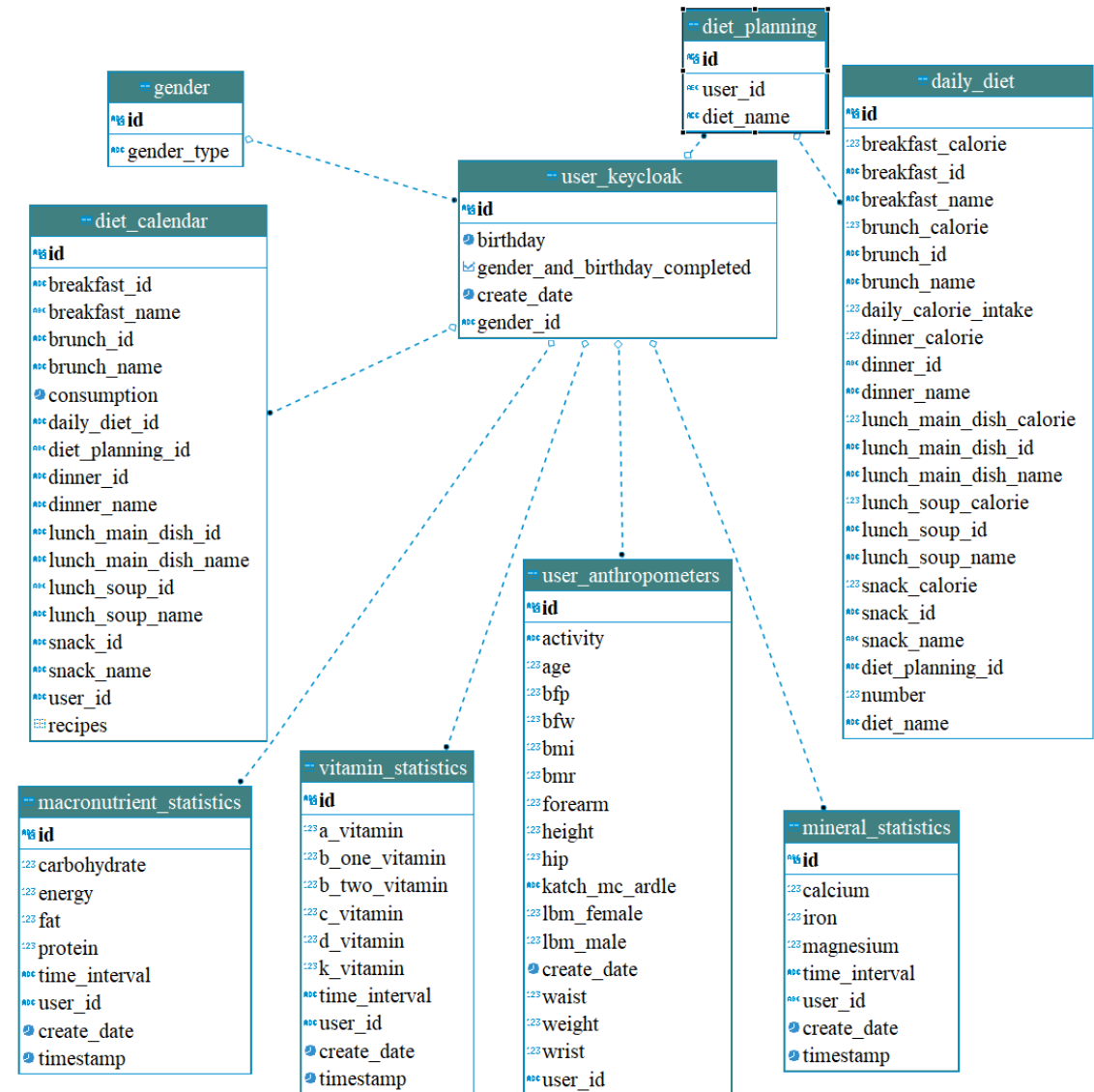
A mértékegység, olaj és cukor preferencia, napszak, árkategória adatai egyértelműek voltak és nem igényeltek nagy erőforrásigényt az adatbázisba történő betöltéséhez. Az élelmiszer adatait az Egészségfejlesztés Iroda [11] weboldaláról töltöttem le és használtam fel. Az adatok felhasználásához Microsoft Excel szoftvert használtam fel. A letöltött táblázatban egyértelmű tagolás utalt arra, hogy milyen típusú élelmiszerrel van dolgom, amelyekhez már csak alkategóriát kellett adnom. Ezeket kigyűjtöttem és generáltam számukra UUID¹⁶ [39] típusú egyedi azonosítót. Az élelmiszereket szét bontottam és átrendeztem az adatbázis táblában lévő oszlopoknak megfelelően, majd elvégeztem az importálást az adatbázisba. A megfelelő egyedi azonosító behelyettesítését FKERES függvény segítségével oldottam meg.

A receptek kidolgozásához segítségül a nosalty weboldalt [40] vettem igénybe. A weboldal alkalmas arra, hogy egy főre is kiszámolja, hogy mennyi alapanyagra van szükség. Hátránya az, hogy sajnos minden elemet és recept leírást manuálisan „kézzel” kellett az Excelbe helyeznem. Ellenben megkönnyítette a munkám abban, hogy valódi és változatos receptekkel tudtam dolgozni. Ahhoz, hogy a recept táblában lévő makrotápanyagok, vitaminok, ásványi anyagok mezőibe megfelelő értékek kerüljenek szükséges volt elvégezni néhány számítást az Excelben. A számítás grammból történő milligrammba átszámítás jelentett és a receptben lévő összes élelmiszer tápanyag értékeit kellett összeadni.

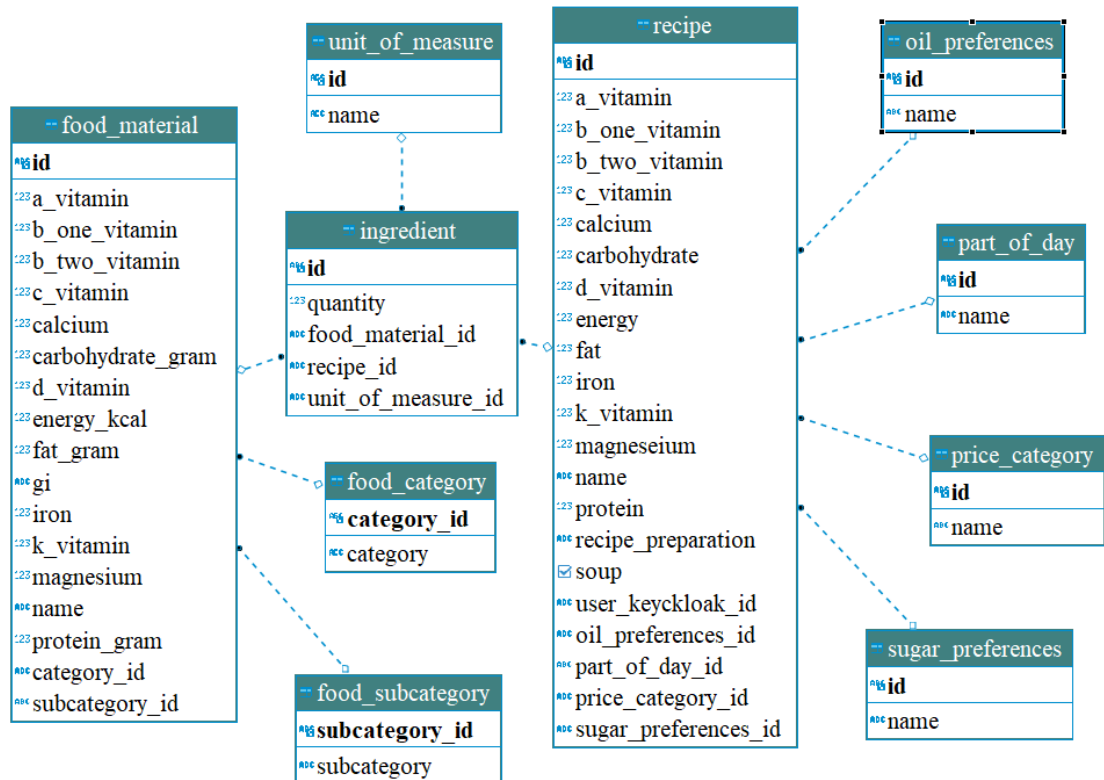
¹⁶ Univerzális egyedi azonosító

	A	B	C	D	E	F	G	H	I	J
1	id	name	part_of_dish_id	protein_id	vit_preferences_id	user_preferences_id	user_keycloak_id	energy_kcal	protein_gm	
2	ca5067b-0a5f-405a-b05c-0a642a27660	Bratkoch krumplás	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		253.894		
3	7e9f952b-080f-4d80-05e0-0b032a2ca343	Kokuszon tejkrém	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		378.25		
4	4001a40f-7e0d-457b-ba0b-70a4d1000010	Franciaosli leves csipke szendvics	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		406.8475		
5	22a52a2f-7605-4a2d-80ef-8a517a7a7070	Árnyas csilis csatlósokrumplás	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		350.6886		
6	5a64a237-8a2c-4a2b-8129-40a01700a4c1	Mázas csilis krumplakrém	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		378.8335		
7	6a552a2f-8144-4770-8409-7165507050e4	Petangkrém	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		318.362		
8	7a0c1a96-8a0f-423e-809f-0a0000000000	Turkós lep	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		493.54		
9	5a461706-5225-461a-9a4d-c8d600a0c03d	Fahéjas almásos	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		386.957		
10	83430a43-5204-46b3-8a05-31a133000000	Árnyas csilis muffin nyírfakrém	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		338.8		
11	200706c2-0506-4a7f-80c5-0706a2a23a7f	Magos csilis krumplakrém	75fa0ee-c072-441f-85b3-879a278b7c21	31022d7b-7038-47e2-bc2d-1a8f64a6a6df	3a2030b-13f0-49e0-939c-07e4de78c279	111a081b-7e05-40e0-803b-7a0722e090		338.8		

8.2. ábra: Összeállított receptek az Excelben adatbázisba történő beimportálásához



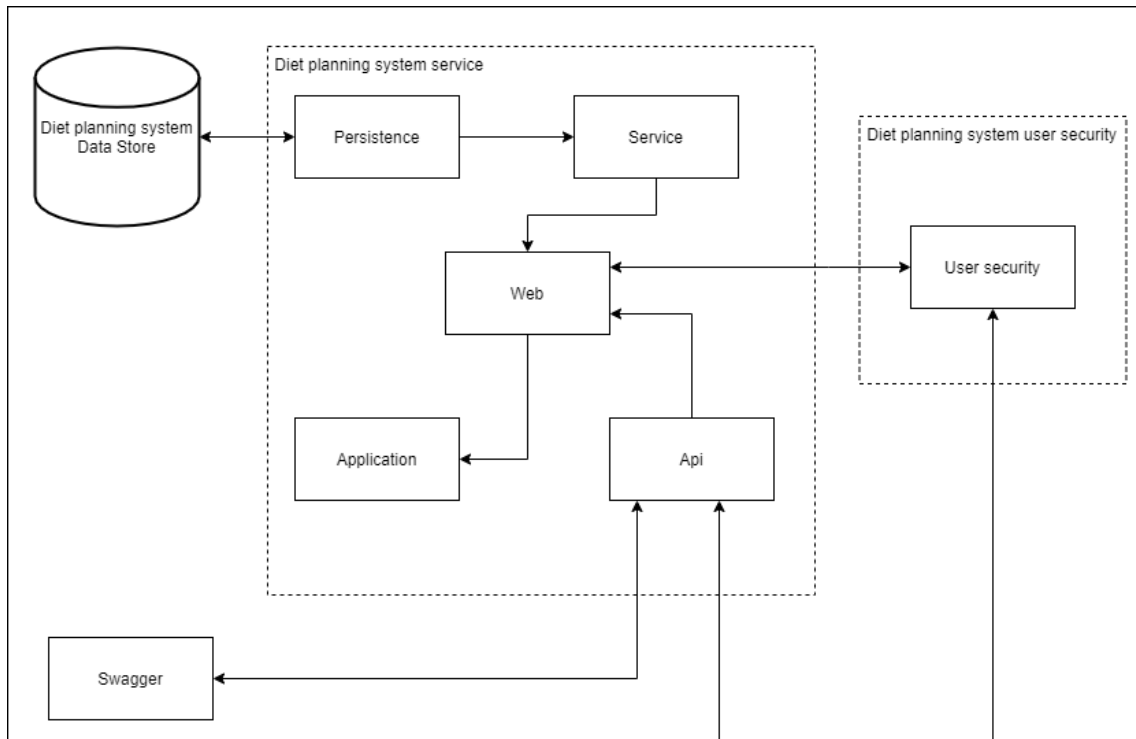
8.3. ábra: Felhasználót érintő táblák



8.4. ábra: Recepteket érintő táblák

8.2 Étrendtervező rendszer szolgáltatás modul

Először elkészítettem egy részletesebb rendszertervet, amely alapján kialakítottam a projektet.



8.5. ábra: Étrendtervező rendszer moduláris felépítése

Az étrendtervező rendszer projekt kialakításához készítettem egy szülő maven projektet. A szülő projekt alá hoztam létre a persistence, web, service, application, api projekteket. Minden projektben hozzáadtam a megfelelő függőségeket, amelyekre szükség volt a szolgáltatás megfelelő futtatásához.

A legfontosabbak:

- Spring Framework
- Springdoc-openapi
- Docker maven plugin
- Springboot WEB
- Springboot data jpa
- PostgreSQL

8.2.1 A legfontosabb maven függőségek összefoglalója [41][42]

A Spring Framework egy Java platform, amely támogatást nyújt Java alkalmazások fejlesztéséhez. A framework kezeli az infrastruktúrát, segítve a fejlesztő munkáját.

A springdoc-openapi segítségével automatikusan létrehozásra kerül a JSON/YAML és HTML alapú dokumentáció, amely elérhetővé válik a swagger-en keresztül. A dokumentáció kiegészíthető megjegyzésekkel. Az én esetemben az alapértelmezett porton érhető el a swagger felület.

A Docker maven plugin segítségével készíthető el fordítás során a docker-image. Amelynek segítségével tudom majd futtatni a projektjeimet docker konténerben.

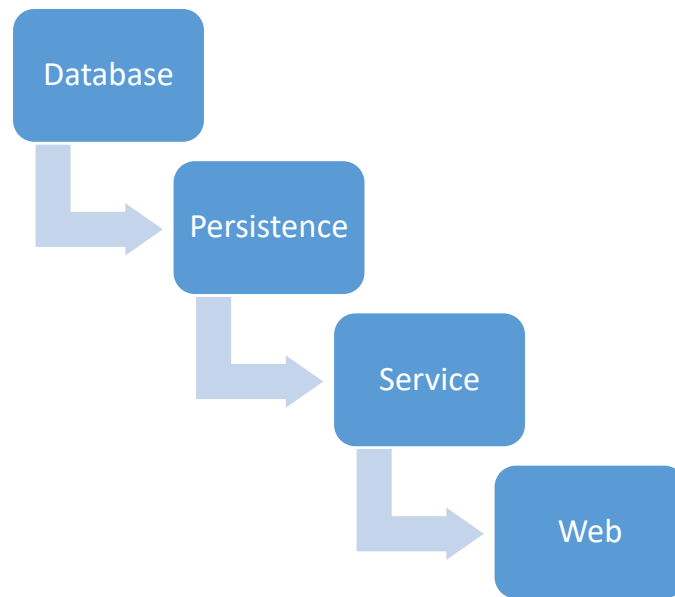
A Springboot WEB függőség segít abban, hogy könnyen létre hozhassak web alkalmazásokat. Létrehozhatok http kiszolgálót, például beágyazott Tomcat használatával.

A Spring Boot data jpa a Spring Data része, amely segítségével könnyebbé teszi a JPA alapú adattárak egyszerűbb megvalósítását.

A PostgreSQL egy JDBC driver. A driver segítségével lehetőség nyílik arra, hogy java kóddal az adatbázisban lévő adatokat lekérdezzük, frissítsük vagy egyéb műveleteket hajtsunk végre.

8.2.2 A felépítés részletezése

A projektek felépítése során ügyeltem arra, hogy a rétegek egymástól teljesen függetlenek legyenek, ugyanis minden rétegnek megvan a saját felelősségköre. A rétegek egymásra épülnek úgy, hogy minden réteg függ az alatta lévőktől. A kapcsolatokat interfészek vagy szervíz típusú osztályok segítségével építettem fel. Ügyeltem arra, hogy az osztályok csak közvetlen hívásokat hajtsanak végre, tehát közvetlenül a web réteg ne hívja meg a perzisztencia réteget.



8.6. ábra: Alkalmazás rétegek elhelyezkedése

Perzisztencia réteg:

Ez felel meg a repository rétegnek. Itt található minden entitás osztály és repository interfész, utóbbi esetben az őse a JpaRepository beépített interfész. Jelen réteg az alkalmazás perzisztencia kezeléséért felelős, amely kommunikál az adatbázissal. A táblák és kapcsolataik felépítését itt valósítottam meg. A felépítés során megadtam, hogy mely mezők milyen típussal rendelkezzenek és milyen kapcsolat jöjjön létre a táblák között. A kapcsolatok tekintetében egy – több kapcsolatokat hoztam létre. A kapcsolatok segítségével könnyen elértem egy adott táblából egy másik tábla adatait, amelyet többször alkalmaztam. Minden entitás osztályhoz készítettem egy interfészt, amely lehetőséget biztosított arra, hogy adatbázisból adatokat tudjak lekérni. A JpaRepository lényege, hogy alapvető primitív írási-olvasási műveleteket biztosítson az adott táblához. Amennyiben az application.properties fájlban a spring.jpa.hibernate.ddl-auto értéke update, create vagy create-drop, akkor az adatbázisban a táblák és azok kapcsolatai automatikusan felépülnek. A megnevezett értékekről még később kitérek.

Service réteg:

A szervíz rétegben jelenik meg az üzleti logika, amely azért felelős, hogy a hozzá érkező paraméterek alapján elvégezze a feldolgozást. Itt történik meg például az adatátviteli objektumok (Data Transfer Object) transzformálása entitás objektumra vagy fordítva. A transzformálás esetében erre egy – egy külön osztályt hoztam létre. Szervíz rétegben lévő egyes osztályokat @Service annotációval láttam el.

Ezeket az osztályokat a Spring Boot be injektálja¹⁷ a kontrollerbe. A Spring Boot a szerviz annotációval ellátott osztályokból egy – egy objektumot állít elő, azaz ezek az osztályok singleton (singleton) tervezési mintát követnek. A beinjektálás azt jelenti, hogy a megfelelő kontrollerben átadjuk konstruktoron keresztül a kívánt szerviz osztályunkat és magát a példány elkészítését és átadását a Spring Boot elvégzi helyettünk.

Ebben a rétegben végeztem el a fontosabb kalkulációkat, amelyek a következők:

- étrend generálása,
- elmúlt időszakban felvett étrendekben szereplő vitaminok, ásványi anyagok stb. eltárolása az adott felhasználóhoz,
- BMI, BMR, LBM stb. kiszámítása,
- naptár kezelése.

Étrend generálása:

Az étrend generálásánál a specifikációban is szereplő elvárásoknak kell megfelelni. Az elvárás az, hogy az adott felhasználó olyan recepteket kapjon eredményül, amely megfelel az általa megadott preferenciáinak. Elvárások tekintetében folyamatos szűrést hajtok végre az adathalmazon, majd elkezdődik az ételek kiválasztása, amely véletlenszerűen történik. Amennyiben kevesebb vagy több a kalória összértéke a felhasználó által elvárt és 10-15% intervallum közötti érték minimumától vagy maximumától, akkor a rendszer kísérletet tesz újabb véletlenszerű ételek kiválasztásával. A sikertelen kiválogatott receptek, tehát az egy – egy adott étrend külön halmazba kerül. Amennyiben minden variáció előállt, akkor ez egy kilépési feltételnek minősül, illetve ebből a halmazból kikerül egy olyan étrend, amely nem felel meg az adott összkalória értéknek, ellenben a legkisebb mértékben tér el az elvárt értéktől.

Elmúlt időszakban felvett étrendekben szereplő vitaminok, ásványi anyagok, makrotápanyagok eltárolása az adott felhasználóhoz:

Amikor a felhasználó belép az oldalra, akkor minden esetben automatikusan elvégzi a rendszer az elmúlt napokban felvett étrendek alapján a vitaminok, ásványi anyagok, makrotápanyagok elmentését a megfelelő táblákban. A rendszer megnézi az étrendeket tartalmazó naptár szerint, hogy mi volt az utolsó bejegyzés azokban a táblákban, ahová elmenti majd az értékeket. Ha van olyan nap, amikor nem történt bejegyzés, akkor a rendszer az adott nap receptjei által összeadja a megfelelő értékeket és elmenti az adott táblába. Például összeszedésre kerül receptenként az összes vitamin értéke és ezek összértéke bekerül a vitamin statisztika táblába. Azért választottam ezt a módszert, mert többnyire sokkal hatékonyabb az ütemezett feladat végrehajtásánál.

¹⁷ Dependency Injection, amely nem GoF tervezési minta. Lényegében az Inversion Of Control egy speciális esete. Röviden egy objektum kívülről kapja meg a használni kívánt objektumot.

BMI, BMR, LBM, BFP kiszámítása:

Az értékek kiszámítása minden esetben akkor történik, amikor egy felhasználó megadja az antropometriai ¹⁸értékeit. A regisztrált és nem regisztrált felhasználók között az a különbség, hogy az előbbi esetben megtörténik az adatbázisba történő mentés, utóbbi esetben pedig nem. A kiszámítás mindkét esetben egyenlő módon zajlik, ellenben a metódusok meghívása különböző api híváson keresztül történik meg. A kiszámításnak a folyamata a „felhasználó képletek” pontnak megfelelően történt.

Naptár kezelése:

A felhasználó számára adottak az általa elmentett étredek. Amennyiben a felhasználó első étrendjét menti el naptárba, akkor az alkalmazás az aktuális naptári napot veszi figyelembe. Egyéb esetben mindig az utolsóként elmentett étrend utolsó napját követően menti el a következő étrend első napi étrendjét. Ezzel megfelel a szoftver az elvárásoknak.

Web réteg:

A web réteg szétszedésre került. Egyik réteg tartalmazza a Rest Controller típusú osztályokat. A másik az API réteg, amely a RequestMapping típusú interfészeket tartalmazza, ugyanitt található az adatátviteli objektumok is. A Web-ben található osztályokba injektálom be a szerviz osztályait és így valósítom meg közvetlen elérést a két réteg között. Az osztályokon továbbá implementálom a RequestMapping típusú interfészeket. Az interfészek metódusain valósítottam meg a post, get, put, delete annotációkat¹⁹. Továbbá ugyanitt határoztam meg azt, hogy milyen api híváson keresztül érhető el az adott funkció, és itt valósítottam meg azt, hogy bizonyos esetekben csak regisztrált felhasználók tudjanak egyes funkciókat elérni.

8.2.3 Swagger

A projekthez hozzáadtam a springdoc-openapi-ui függőséget. Miután ez megtörtént, elvégeztem a szükséges dokumentálást az API interfészekben, amely automatikusan megjelent a Swagger felületén is a springdoc-openapi segítségével. A Swagger az alapértelmezett 8080-as porton a swagger-ui/index.html?configUrl=/v3/api-docs/swagger-config keresztül érhető el.

¹⁸ Antropometria egy tudomány, amely az emberi test részeinek mérésével és méretével foglalkozik.

¹⁹ Olyan nyelvi elemek, amelyek lehetővé teszik a metaadatok beépítését a forráskódba.

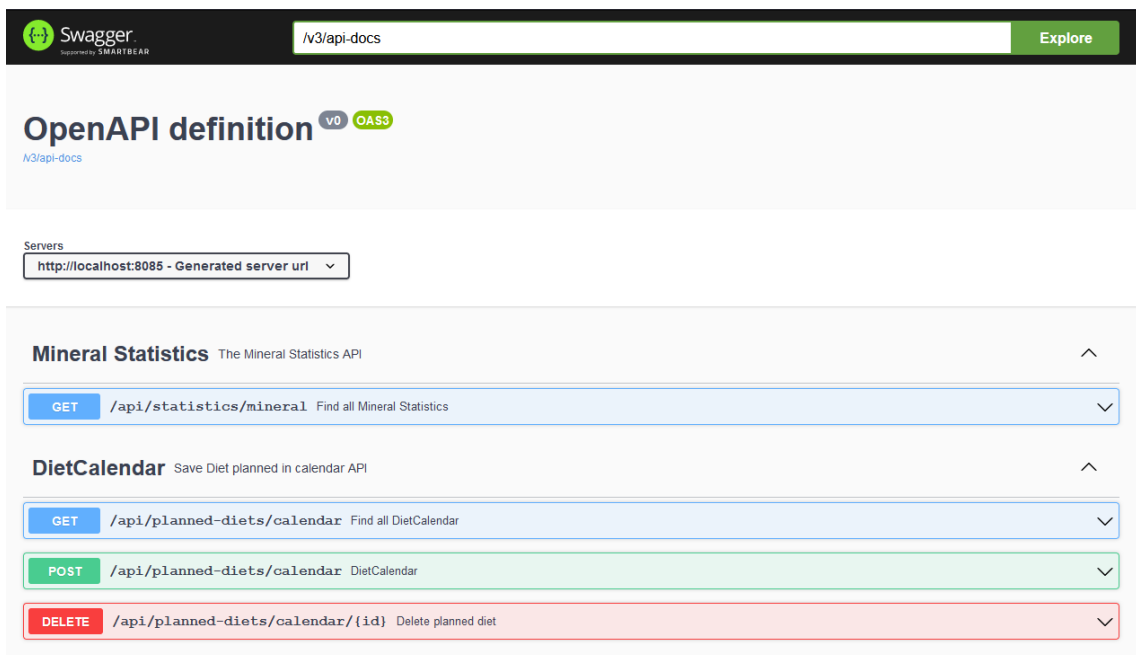
```

20 @RequestMapping(value = "/api")
21 @Tag(name = "DietCalendar", description = "Save Diet planned in calendar API")
22 public interface DietCalendarApi {
23
24     @Operation(summary = "DietCalendar", description = "Save Diet planned in calendar", tags = { "DietCalendar" })
25     @ApiResponse(responseCode = "200", description = "Successful operation.")
26     @PreAuthorize("hasRole('REGISTERED_USER')")
27     @PostMapping(value = "/planned-diets/calendar", consumes = MediaType.APPLICATION_JSON_VALUE, produces = MediaType.APPLICATION_JSON_VALUE)
28     void save(@RequestBody List<DietCalendar> dietCalendar, Authentication anAuthentication);
29
30     @Operation(summary = "Find all DietCalendar", description = "Returns all DietCalendar", tags = { "DietCalendar" })
31     @ApiResponse(responseCode = "200", description = "Successful operation.")
32     @PreAuthorize("hasRole('REGISTERED_USER')")
33     @GetMapping(value = "/planned-diets/calendar", produces = MediaType.APPLICATION_JSON_VALUE)
34     List<DietCalendar> getAll(Authentication anAuthentication);
35
36     @Operation(summary = "Delete planned diet", description = "Returns success OK", tags = { "DietCalendar" })
37     @ApiResponse(responseCode = "200", description = "Successful operation.")
38     @PreAuthorize("hasRole('REGISTERED_USER')")
39     @DeleteMapping(value = "/planned-diets/calendar/{id}")
40     void delete(@PathVariable("id") String id, Authentication anAuthentication);
41 }

```

8.7. ábra: Étrend naptár API interface

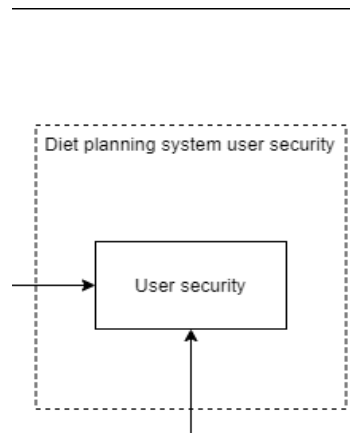
Ahogy a 8.4-es ábrán is látható mindent, amelyet a @Tag, @Operation, @ApiResponse és @PostMapping annotációban lévő információ olvasható az megjelenítésre kerül a Swagger felületén is. Ahogy a következő 8.5-ös ábrán is látható.



8.8. ábra: Swagger webes felülete

8.3 Étrendtervező rendszer felhasználó hitelesítését kezelő modul [43]

A felhasználók hitelesítés végrehajtására egy új projektet hoztam létre.



8.9. ábra: Felhasználó hitelesítés modul

A projekt nem szolgáltatásként, hanem modulként működik. Pontosabban a modult, egy másik projekt tudja használni és külön nem futtatható. Az én esetemben az étrend tervező szolgáltatás projekt használja ezt a modult. Feladata az, hogy a Keycloak-al kommunikáljon. A modulba beillesztettem az egyik legfontosabb függőséget a keycloak-spring-security-adatptert. Segítségével ellenőrizhetem, a megkapott Token alapján, hogy ki jogosult egy-egy adott funkció elérésére és ki nem. Amennyiben sikeres a hitelesítés, akkor ebben az esetben az adott API híváson keresztül végrehajtásra kerül a megfelelő műveletek sorozata. Ha nem, akkor hibaüzenet kerül visszaküldésre a kliens felé.

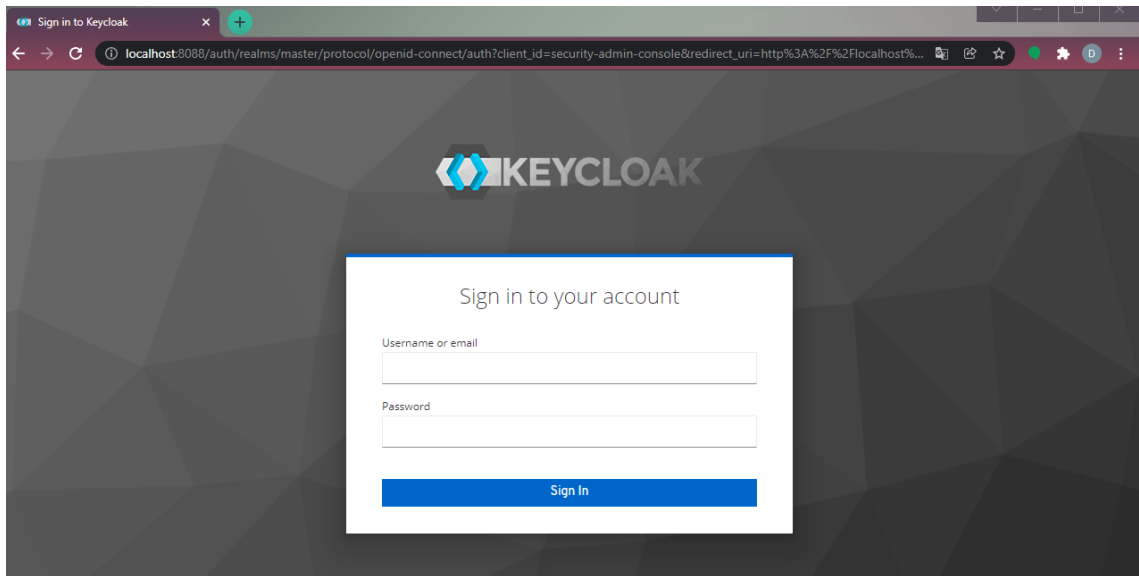
8.4 Docker [45][46]

A Docker Desktop-ot letöltöttem és telepítettem a számítógémemre. Ezt követően létrehoztam egy docker-compose.yml fájlt. A docker compose egy többtárolós Docker alkalmazások definiálására és futtatására alkalmas eszköz. A yaml fájlban elvégezhető a szükséges konfiguráció, amelyben implementálhatók²⁰ az alkalmazások. A beállítások elvégzését követően egy paranccsal elindítható az összes szolgáltatás. A PostgreSQL-t, a Keycloak-ot, étrendtervező rendszer felhasználó hitelesítését kezelő modult és az étrendtervező rendszer szolgáltatás konfigurációit végeztem el a docker-compose fájlban. A konfigurációban megadtam a bemenő és kimenő portok értékét, amelyeken a szolgáltatások elérhetőek a Docker elindítását követően.

²⁰ Megvalósítás

8.5 Keycloak [47]

Ebben a fejezetben bemutatom, hogyan végeztem el a Keycloak server konfigurációt. Az adminisztrációs webes felület a dockerben megadott porton keresztül könnyen elérhető.



8.10. ábra: Adminisztrátor bejelentkezési felülete

Az adminisztrációs felületre belépve hoztam létre az új klienst, amely a dps (Diet planning system) nevet kapta, hogy könnyen beazonosítható legyen a későbbi belépések során.

Dps

Settings Keys Roles Client Scopes Mappers Scope Revocation Sessions

Client ID

Name

Description

Enabled ☒

Always Display in Console ☐

Consent Required ☐

Login Theme

Client Protocol

Access Type

Standard Flow Enabled ☒

Implicit Flow Enabled ☐

Direct Access Grants Enabled ☒

OAuth 2.0 Device Authorization Grant Enabled ☒

Root URL

* Valid Redirect URIs

Base URL

Admin URL

Web Origins

Backchannel Logout URL

Backchannel Logout Session Required ☒

Backchannel Logout Revoke Offline Sessions ☐

8.11. ábra: Kliens beállítása

Ugyanitt hoztam létre egy új szabályt, amelynek a neve a regisztrált felhasználó. A szabály segítségével korlátozom, az alkalmazásom egyes funkcióinak elérhetőségét, a nem regisztrált látogatók számára.

Keycloak Admin Console

localhost:8080/auth/admin/master/console/#/realms/dps/clients/4f4cecc0-f864-4f25-8cb5-8f03a29786f0/roles

KEYCLOAK

Dps

Configure

Realms Settings

Clients

Client Scopes

Roles

Identity Providers

Clients > dps

Dps

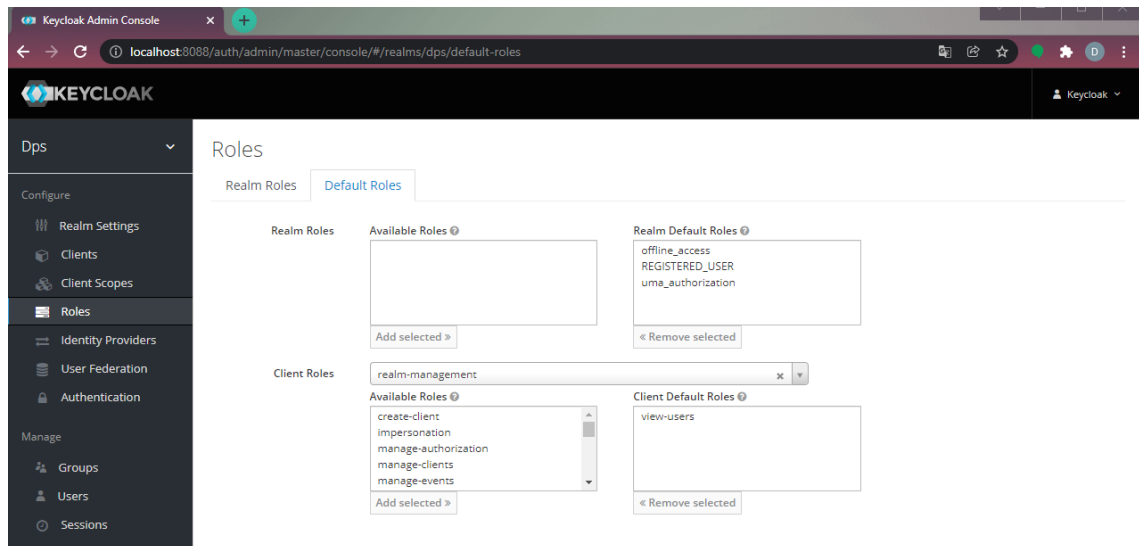
Settings Keys Roles Client Scopes Mappers Scope Revocation Sessions Offline Access Installation

Search... View all roles Add Role

Role Name	Composite	Description	Actions
REGISTERED_USER	False		Edit Delete
uma_protection	False		Edit Delete

8.12. ábra: Szabály létrehozása

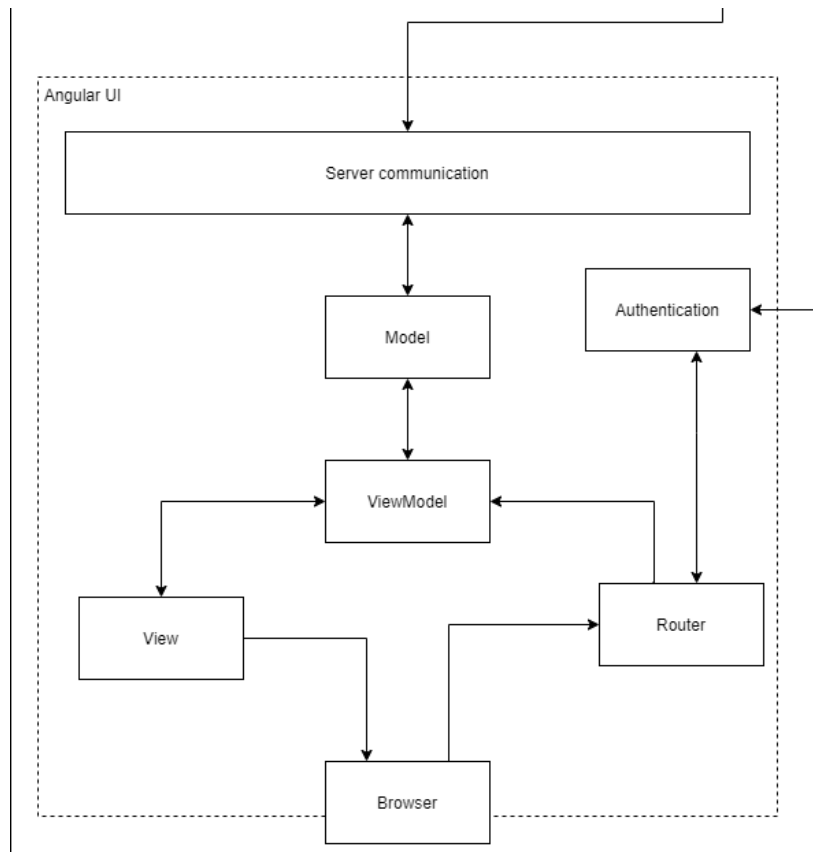
Az API projektben lévő API hívásokért felelős interfészekben adtam meg azoknál a függvényeknél a fentebb említett szabály elnevezést, ahol azt szerettem volna elérni, hogy ezek az API hívások csak azon felhasználók számára legyenek elérhetőek, akik jogosultak rá. A következő lépésként az újonnan felvett szabályt hozzáadtam alapértelmezett szabályként, tehát, amikor egy új felhasználó regisztrál a Keycloak rendszeren keresztül, akkor az általam felvett szabály is hozzáadásra kerül az új felhasználóhoz.



8.13. ábra: Általam létrehozott szabály alapértelmezetté állítás

8.6 Angular applikáció

Amikor elkészült a teljes modellezés és tervezés, akkor következő lépésként meg kellett határozni, hogy milyen kialakítása legyen a teljes weboldalnak. A célom az volt, hogy könnyen kezelhető és áttekinthető felület készüljön el. A megoldást a Material Design modulban láttam. [48] A modul abban nyújt segítséget, hogy nem a fejlesztőnek kell felépítenie az alapoktól a felületi elemeket (komponenseket). Az idő rövidege végett, csak minimális változtatásokat eszközöltem az egyes modulokon, ellenben a végeredményként egy letisztult, egységes és reszponzív felületet kaptam. A hivatalos oldalon pontos leírás érhető el a komponensekről és minták alapján végeztem el az elemek beintegrálását a saját alkalmazásomba. Az alkalmazást a részletes rendszerterv alapján fejlesztettem. A UI template, autentikáció és statisztika chartok npm package-ként lettek betöltve az applikációba és ezeken hajtottam végre módosításokat, finomítottam úgy, hogy az megfeleljen az elvárt működésnek.



8.14. ábra: Anuglar alkalmazás moduláris felépítése

8.6.1 Útvonalválasztó

Az útvonalválasztóban azt határoztam meg, hogy egyes webcímek esetén, milyen nézet jelenjen meg. Feladata továbbá az is, hogy regisztráció és felhasználói belépés esetén elirányítson a megfelelő Keycloak felületre a hitelesítési rétegen keresztül. Ebben a rétegben valósítottam meg azt is, hogy egyes webcímek esetén bejelentkezéshez kötöttem a nézet elérését. Amennyiben egy adott felhasználó egy olyan oldalt kísérel meg elérni, amelyre a bejelentkezett felhasználó szabály vonatkozik, akkor automatikusan visszairányít a rendszer a kezdőlapra.

8.6.2 Hitelesítés [49][50]

A hitelesítés során néhány konfigurációt kellett elvégezni a Keycloak szerveren. Ahhoz, hogy megfelelő legyen a kommunikáció a gyökér webcímet és a kliens protokollt kellett beállítani. Előbbi esetben egy portot állítottam, utóbbi esetben openid-connect típust kellett kiválasztani. Angular alkalmazásomban megadtam a webcím, realm és clientid értékét, amelyek a Keycloak adminisztrátor oldalán elérhetők. A réteg felelőssége a felhasználó elirányítása a Keycloak regisztrációs vagy bejelentkezési felületéhez.

Magát a hitelesítést a Keycloak szerver végzi el. További felelőssége a rétegnek az, hogy ha a felhasználó olyan webcímet próbál meg elérni, amely bejelentkezéshez kötött, akkor elvégzi annak az ellenőrzését, hogy token alapján a felhasználó bejelentkezett állapotban van vagy sem.

8.6.3 Nézet

A nézet rétegben valósítottam meg azokat a komponenseket, amelyek kifejezetten a megjelenítésekért felelnek. Minden komponens esetén különválasztottam az egyes elemeket. Ezáltal a komponensek újra felhasználhatók váltak. Ilyen például a recept vagy statisztikai elemek megjelenítése. Az adatokat a model réteg objektumaiból kapja a nézet – model rétegen keresztül.

8.6.4 Nézet – model

A nézet – model rétegbe került az üzleti logika. Az Angular alkalmazásban lévő üzleti logika nem ekvivalens²¹ a szerver oldal üzleti logikájával. Az alkalmazás a szerver oldaltól úgy kapja meg az adatokat, ahogyan megjelenik a felületen. Az adatok megjelenítésének módját változtatja meg a réteg. Jelen rétegben valósul meg az eseményekre történő feliratkozás is, amelynek lényege, ha valamilyen változás történik az adaton, akkor erről értesítést kapjon ez a réteg.

8.6.5 Model

Ebben a rétegben csupán az adatátviteli objektumok osztályai jelennek meg. Az adatátviteli objektumok szerkezete ekvivalens a szerver oldali adatátviteli objektumaival.

8.6.6 Szerver kommunikáció

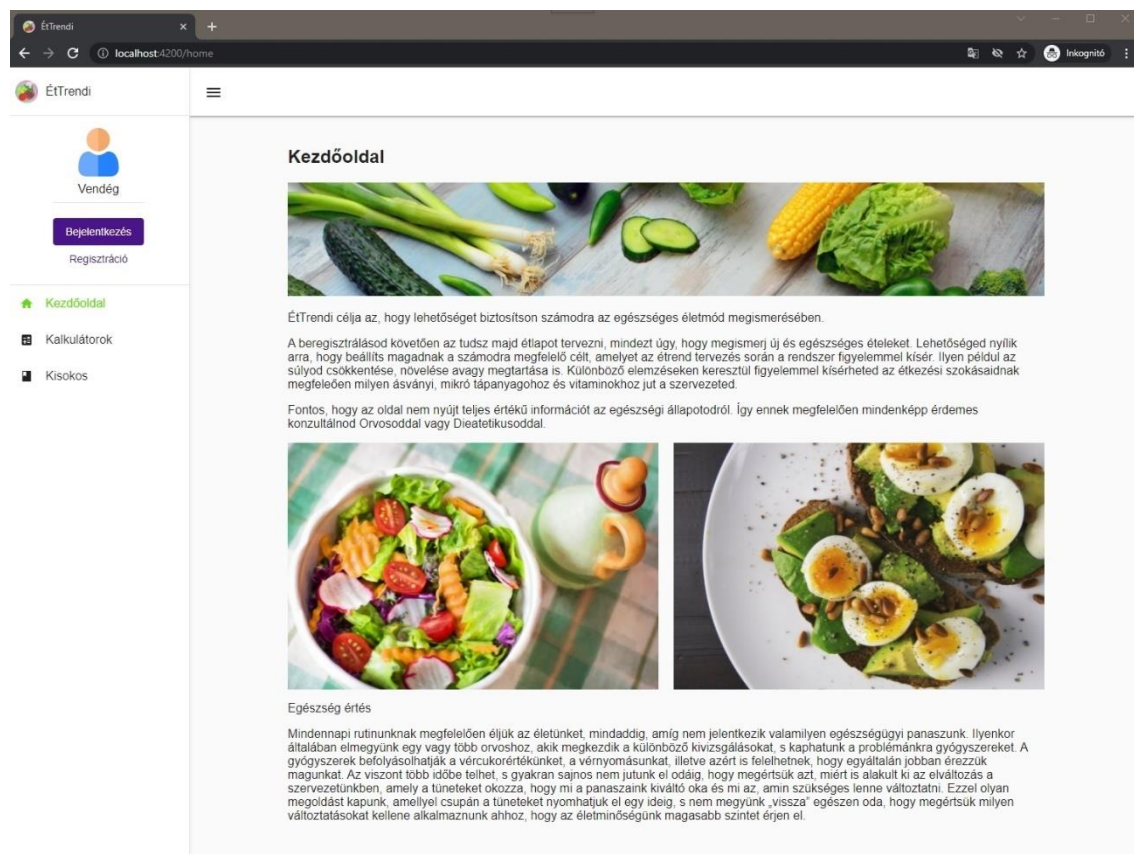
A szerver kommunikáció rétegben történnek az API hívások. Ez a réteg áll kapcsolatban API-n keresztül a szerver oldallal és adja vissza a nézet – model felé a szervertől kapott adatokat a modelben található adatátviteli objektumok segítségével.

²¹ Egyenlő értékű

8.7 Alkalmazás bemutatása

8.7.1 Főoldal

A főoldalon látható elemek elhelyezkedése jellemzi az egész weboldalt. Jobb oldalon látható a menü és bejelentkezés – regisztráció, középen egy rövid ismertető.



8.15. ábra: A főoldal

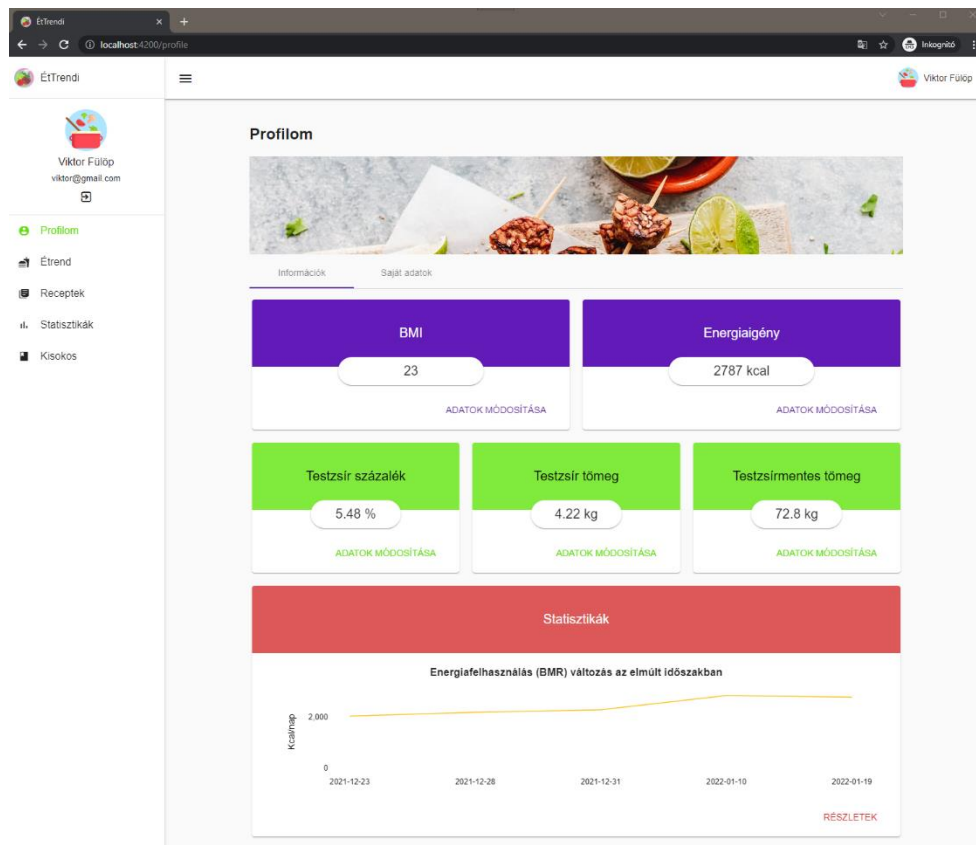
A baloldali menük által érhetőek el a kezdőlap, ahogyan a fenti képen is látható, a kalkulátor és a kisokos. A következő képen az Energiaigény kalkulátor látható a beírt paraméterekkel és kiszámolt értékekkel. A kalkulátorok eléréseit harmonika típusú elemmel oldottam meg, emiatt a képen látható elemek között az aktív kalkulátor látszik.

8.16. ábra: Energiaigény kalkulátor

8.7.2 Felhasználói kezelőfelület

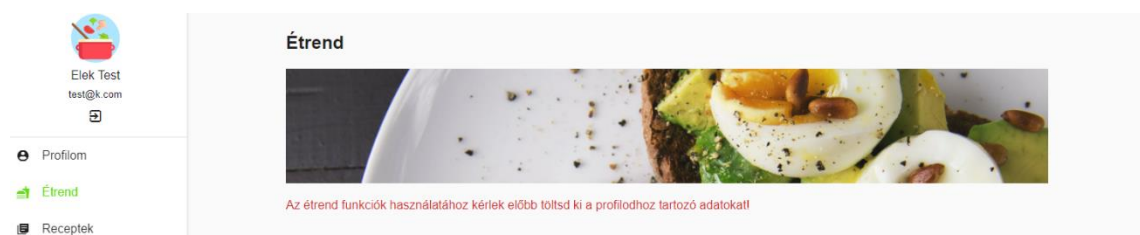
Amikor a webalkalmazásba történő belépés bekövetkezik, akkor elsőként a személyes kezelőfelületet fogja látni a felhasználó. Itt megtekintheti azokat az eredményeket, amelyek az antropometriai értékekből történő számítás eredményei. Továbbá elérhető az energiaigény statisztikája az elmúlt 30 nap tekintetében. Amikor a felhasználó regisztráció után végrehajtja az első belépést, akkor az elemek látszódnak, ellenben értékeket nem tartalmaznak. Kivételt képez az energiaigényt mutató statisztika, mert az említett esetben ez az elem nem érhető el.

8.17. ábra: Elemek az első belépést követően



8.18. ábra: Már kitöltött antropometriai értékek által látható eredmények

Amikor a felhasználó elsőként lép be az oldalra, akkor a saját adatok kitöltése nélkül nem engedélyezi az alkalmazás az étrend generálását.



8.19. ábra: Nem elérhető funkció

Ez azért fontos lépés, mert például az energiaigény fontos szerepet játszik abban, hogy milyen ételeket válasszon ki az étrendhez a kalkuláció során a szoftver. A saját adatok kitöltésénél a nem, és a születési dátum kitöltése egyszer lehetséges. Az adatok elmentése során a rendszer megerősítést vár az adatok elmentése előtt. Miután megtörténik a mentés, akkor az említett értékek láthatók, ellenben már nem módosíthatóak.

Profilom

Információk Saját adatok

Saját adatok
Ezek az adatok szükségesek a BMI és a saját energiaigény kiszámításához.

Nem: Férfi Születési dátum: 1985-09-30

Magasság (cm): Testsúly (kg): Derék kerülete (cm):

Figyelem!
A "Nem" és a "Születési dátum" mezők a későbbiekben már nem változtathatók. Biztosan elmented így az adatokat?

Nem Igen

8.20. ábra: Saját adatok megadása

Egyetlen nap csak egyszer adható meg minden anropometriai érték a specifikációnak megfelelően. Mielőtt elmentésre kerülnek az értékek, akkor szintén megerősítést vár az alkalmazás. Amennyiben sikeres volt a mentés, akkor a rendszer értesíti a felhasználót arról, hogy másnap tud majd újra értékeket felvenni.

Saját adatok
Ezek az adatok szükségesek a BMI és a saját energiaigény kiszámításához.

Nem: Férfi Születési dátum: 1985-09-30

Magasság (cm): 170 Testsúly (kg): 60 Derék kerülete (cm): 50

Aktivitás: Mérsékeltén aktív életmód (heti 1 – 3 nap könnyed edzés)

Az adatok változtatása egy nap csak egyszer lehetséges. Kérlek várj holnapig!

8.21. ábra: Kitöltött felhasználói adatok

8.7.3 Étrend

Elsőként étrendet vagy étrendeket kell generáltatnom a rendszerrel. Ennek során a specifikációban elvártak szerint sorrendben kiválasztható a kívánt elem, ilyen például, hogy milyen típusú olajok jelenjenek meg a receptekben.

Étrend

Étrend ütemezés Étrend tervezés Étrendjeim

Új étrended megtervezéséhez kérlek válaszold meg az alábbi kérdéseket, amely csak pár percet fog igénybe venni!

1. Mi a célod ezzel az étrenddel? **Fogyás**

2. Milyen árkategóriába essen az étrend? **Olcsó**

3. Van-e egészséges olajokkal kapcsolatos preferenciád? **Olívaolaj**

☐ Nincs

☐ Avokádóolaj

☒ Olívaolaj

☐ Dióolaj

☐ Repceolaj

☐ Lenmagolaj

4. Van-e cukorral kapcsolatos preferenciád?

5. Van-e étkezésekkel kapcsolatos preferenciád?

6. Hány napra tervezed az étrendet?

7. Véglegesítés és tervezés

8.22. ábra: Étrend tervezés folyamata

Amikor elkészült a generálás, akkor a szoftver felajánl egy étrendet annyi napra, amennyit meghatározott a felhasználó. Amennyiben a felhasználó úgy dönt, hogy el szeretné menteni, akkor abban az esetben elnevezi az étrendjét és elmenti. Ha nem találja megfelelőnek az ajánlatot, akkor ez esetben elvégezheti egy kattintással az újra tervezést a már beállított értékekkel. Arra is lehetőséget biztosít a felület, hogy a felhasználó más beállítások mellett végezze el a tervezést. A tervezett étrendendeken látszik a teljes napi és napszaknak megfelelő kalória mennyiség összeadott értéke. Összeadott kalória érték alatt azt értem, amikor egy recept több élelmi anyag kalória értéke, illetve a napszakok receptjeinek a kalória értéke kerül összeadásra. Továbbá látható a napszak megnevezése és az, hogy milyen receptet kínál a rendszer.

Készen is vagyunk, mindössze egy kattintásra vagy személyre szabott étrendedtől!

Nevezd el az étrended! *

Étrend mentése **Étrend újratervézése** [Kezdjük előlről](#)

1. nap (2,101 kcal)	2. nap (2,448 kcal)	3. nap (2,714 kcal)	4. nap (2,223 kcal)
Reggeli: Tükörtojás (195.65 kcal)	Reggeli: Hagymás-kolbászos tojásrántotta (538.8 kcal)	Reggeli: Vajas kalács mézzel (756.6 kcal)	Reggeli: Amerikai palacsinta (309.16 kcal)
Ebéd: Leves: Frankfurti leves szója virslivel (456.89 kcal) Főétel: Rakott krumpli (661.03 kcal)	Ebéd: Leves: Brokkoli krémleves (252.49 kcal) Főétel: Borsó főzelék (382.15 kcal)	Ebéd: Leves: Fokhagyma krémleves (277.95 kcal) Főétel: Zöldséges rakottas (335.44 kcal)	Ebéd: Leves: Fahéjas almaleves (166.96 kcal) Főétel: Rakott krumpli (661.03 kcal)
Uzsonna: Kakaós csiga (501.48 kcal)	Uzsonna: Kókuszgolyó kávéval (626.4 kcal)	Uzsonna: Leveles kókuszos csiga (684.46 kcal)	Uzsonna: Csokis muffin (591.94 kcal)
Vacsora: Csilis bab húsmintesen (286.38 kcal)	Vacsora: Paprikás krumpli (648.19 kcal)	Vacsora: Pirítós sonkával (659.84 kcal)	Vacsora: Tortilla lap (493.54 kcal)

8.23. ábra: Generált nem mentett étrend

Az elmentett étrendek megtekinthetők az étrendjeim menüpont alatt, ahol látható, hogy az adott recept milyen összetevőket tartalmaz. Az étrendek nevek alapján beazonosíthatóak, amelyeket a felhasználó maga ad meg annak elmentésekor. A felhasználó megteheti, hogy az adott étrendjét kitörli, vagy hozzáadja a naptárához. Amennyiben egy adott étrend már szerepel a naptárban, akkor is kitörölheti az étrendet a felhasználó. A naptárban azonban ekkor sem kerül eltávolításra, mert ebben az esetben a statisztikai adatok sérülnek.

Étrendieim

17

Étrend törlése

Az étrendek naptárba történő elhelyezése az étrend ütemezés menüpont alatt található. A kiválasztott étrendet egyszerűen csak hozzáadom a naptárhoz. A rendszer automatikusan vagy a meglévő étrend után egy nappal helyezi be az étrendet sorrendben, vagy ha az utolsó étrend régebbi, mint az aktuális nap, akkor a rendszer az aktuális napot veszi figyelembe. Utóbbi eset akkor is igaz, ha még nem szerepel a naptárban étrend. A naptárban elérhető étrendek receptjei elérhetővé válnak ezáltal látható a receptek tartalma és leírása.

Étrend ütemezés

Étrend tervezés

Étrendjeim

Választható étlendek

Étlend hozzáadása

2021-12-22

Reggeli - Lágytojás rozskenyérrel

Ebéd (Leves) - Fahéjas almaleves

Ebéd (Főétel) - Burgonya főzelék

Uzsonna - Kókuszgolyó túróból

Vacsora - Tortilla lap

Receptek megtekintése

2021-12-23

Reggeli - Bundás kenyér

Ebéd (Leves) - Hagymaleves sajtos pirítóssal

Ebéd (Főétel) - Sárgabab főzelék

Uzsonna - Almás pite

Vacsora - Paprikás krumpli

Receptek megtekintése

2021-12-24

Reggeli - Zabkása

Ebéd (Leves) - Fahéjas almaleves

Ebéd (Főétel) - Rakott krumpli

Uzsonna - Almakompót

Vacsora - Tortilla lap

Receptek megtekintése

2021-12-25

Reggeli - Lágytojás rozskenyérrel

Tízórai - Gyömbéres tea

Ebéd (Leves) - Brokkoli krémleves

Ebéd (Főétel) - Sárgaborsó főzelék

Uzsonna - Palacsinta

Vacsora - Paprikás krumpli

Receptek megtekintése

2021-12-26

Reggeli - Bundás kenyér

Tízórai - Banán

Ebéd (Leves) - Fokhagyma krémleves

Ebéd (Főétel) - Zöldséges rakottas

Uzsonna - Sajtos pogácsa

Vacsora - Tortilla lap

Receptek megtekintése

2021-12-27

Reggeli - Hagymás-kolbászos tojásrántotta

Tízórai - Banán

Ebéd (Leves) - Hagymaleves sajtos pirítóssal

Ebéd (Főétel) - Sárgaborsó főzelék

Uzsonna - Leveles kókuszos csiga

Vacsora - Tortilla lap

Receptek megtekintése

2021-12-28

Reggeli - Lágytojás rozskenyérrel

Tízórai - Banán

Ebéd (Leves) - Brokkoli krémleves

Ebéd (Főétel) - Sárgabab főzelék

Uzsonna - Sajtos pogácsa

Vacsora - Paprikás krumpli

Receptek megtekintése

2021-12-29

Reggeli - Rozsos kenyér felvágottal zöldségekkel

Tízórai - Alma

Ebéd (Leves) - Fokhagyma krémleves

Ebéd (Főétel) - Sárgabab főzelék

Uzsonna - Almakompót

Vacsora - Paprikás krumpli

Receptek megtekintése

8.25. ábra: Étlend naptár

Étrend receptek

[Vissza az étrendhez](#)

Reggeli

Lágytojás rozskenyérrel (661 kcal)

Leves

Fahéjas almaleves (167 kcal)

Hozzávalók

- Pudingpor 10 dkg
- Alma 5 gramm
- Tehéntej (1.5%-os) 3 db
- Fahéj 50 gramm
- Szegfűszeg (őrölt) 0.5 db
- Eritrit 1 ml
- Víz 6 dkg

Elkészítés

Három darab almát meghámozunk, apró kockákra vágjuk, majd egy lábasba helyezzük, és felöntjük 3dl vízzel. 5 percre forraljuk, közben ízesítjük fahéjjal, szegfűszeggel. Amíg összeformak az ízek, előkészítjük a pudingot, mellyel sűríteni fogjuk. A pudinghoz teszünk 5 gramm eritritet, majd felöntjük 1dl tejjel, és csomómentesre keverjük. Ezután ezt folyamatos keverés közben hozzáöntjük az almalevesünkhöz, felforraljuk, és amikor már sűrűsödik, készen is van, tálalhatjuk.

Főétel

Burgonya főzelék (236 kcal)

Uzsonna

Kókuszgolyó túróból (428 kcal)

Vacsora

Tortilla lap (494 kcal)

8.26. ábra: Részletes recept leírások

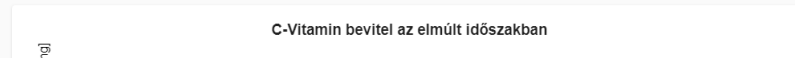
8.7.4 Receptek

A rendszerben lévő receptek mindegyike elérhető a felületen, ellenben a felhasználók számára ezek csak olvasható formátumban érhetők el. Ennek oka, hogy ha változtat a felhasználó a receptek tulajdonságain, akkor csorbul annak hitelessége, hogy milyen eredményeket mutat majd a statisztikában.

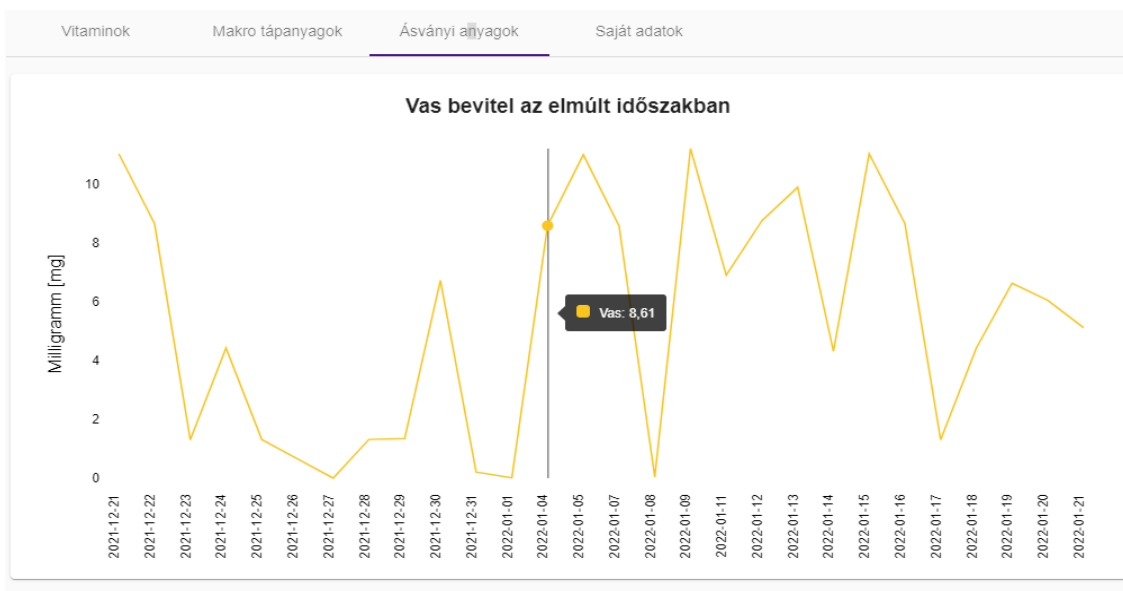
8.7.5 Statisztika

A statisztikai adatoknál minden vitamin, makrotápanyagok, ásványi anyagok és saját adatok értékei láthatóak visszamenőleg. A felhasználó pontosan látja, hogy milyen értékű - a már felsorolt - anyagok és vitaminok kerülnek a szervezésébe. Továbbá látja, hogyan változnak a testtömeg-index, energiaigény, testzsír százalék és testzsír tömeg értékei.

Saját adatok



8.27. ábra: Vitaminok statisztikai adatai



8.28. ábra: Ásványi anyagok vas statisztikai adata

9 Tesztelés

9.1.1 Végétől a végéig tartó tesztelés – E2E

A tesztelés során megvizsgáltam a nem regisztrált, illetve regisztrált és bejelentkezett felhasználók által elérhető funkciókat.

- Nem regisztrált felhasználók esetén:
 - A kalkulátor elérésével és adatok megadásával eredményre jutás.
 - A kalkulátor elérésével és helytelen adatok megadásával nem jut eredményre a látogató.
 - Profilom, étrendek, receptek, statisztikák nem érhetők el a nem bejelentkezett felhasználók esetén.
- Regisztrált és bejelentkezett felhasználó:
 - Profilom, étrendek, receptek, statisztikák csak akkor érhetők el, ha a bejelentkezés megtörtént.
 - Első bejelentkezést követően csak akkor végezhet étrend generálást a felhasználó, amennyiben kitöltötte a saját adatait.
 - A saját adatok kitöltése során csupán egyszeri lehetősége van a felhasználónak a nem, és születési idő kitöltésére. A kitöltést követően a megadott adat látható ellenben nem konfigurálható.
 - A saját adatok menüpontban lévő antropometriai értékek naponta csak egyszer adhatók meg.
 - A saját adatok menüpontban lévő mezők kitöltése során helytelen adatok megadása. A mentés során hibaüzenet érkezik a felhasználó felé.
 - A saját adatok kitöltését és mentését követően az információk menüben lévő adatok elérhetővé válnak.
 - Étrend ütemezés során étrendet próbál a felhasználó úgy hozzáadni, hogy nem választja ki az étrendet.
 - Étrend ütemezés során étrendet próbál a felhasználó úgy hozzáadni, hogy kiválasztja az étrendet. Majd az étrend receptjei megjelennek a naptárban.
 - A naptárban lévő receptek részletes információi megtekinthetők a felhasználó számára.
 - Étrend tervezés során a felhasználó megpróbál úgy étrendet generálni, hogy nem tölt ki minden mezőt. Ebben az esetben nem történhet meg a generálás.
 - Étrend tervezés során a felhasználó megpróbál úgy étrendet generálni, hogy minden mezőt kitölt, majd megjelenik a szoftver által kínált étrend annyi napra, amennyit beállított. Az étrendekben látszódik a kalória, a napszak és az étel neve.
 - Az étrend újra generálása során a már beállított értékek nem változnak.

- Az étrend mentése csak akkor lehetséges, amennyiben a felhasználó elnevezte azt. Az étrend megjelenik az étrendjeim menüpont alatt.
- Az étrendjeim menüpontban látható az összes étrend napokra leosztott receptjei.
- Az étrendjeim menüpontban az étrend törlése megtörténik anélkül, hogy elveszne a már naptárban szereplő adott étrend.
- Az étrend menüpontban lévő elem hozzáadható a naptárhoz.
- A statisztikai adatokban lévő vitaminok, makrotápanyagok, ásványi anyagok és saját adatok grafikonjai látszódnak és értelmezhetőek.

A tesztekéről tesztforgatókönyvet készítettem egy Excel fájlban. Minden tesztet, tesztlépésekre bontottam és kiértékeltem azokat. Minden esetben rögzítettem, hogy az adott tesztlépés során milyen tesztadattal dolgoztam. A tesztforgatókönyvet úgy építettem fel, hogy az eredmény kiértékelésekor összevettem a rendszer követelmény specifikációjával.

9.1.2 Unit teszt

Szerver oldalon lévő üzleti logika rétegben végeztem unit tesztelést. Csak a komplexebb függvényeket tartalmazó osztályok tekintetében írtam Unit tesztet. Ilyen például az étrend generálása vagy a felhasználó energiaigény, testtömeg-index stb. kalkulációja. Az említett osztályok tesztelését TDD módszerrel hajtottam végre. Először felmértem a feladatot, majd elkészítettem a tesztelendő osztályt, majd ezután következett a teszt osztály elkészítése. Azoknál a függvényeknél a fejlesztés viszonylag gyorsan történt, amelyeknél kész megoldásaim voltak. Ilyen például a képletek, amelyeket például az energiaigény kiszámításánál alkalmaztam.

Nem törekedtem arra, hogy 100%-os lefedettségem legyen a tesztelés során, ugyanis a legtöbb esetben a függvények minimális logikát tartalmaznak. A Unit tesztelés nagyon hasznos, mivel segít rávezetni a tiszta kód alapelvekre, ellenben például a TDD módszer esetében a fejlesztési idő megduplázódhat.

10 Tovább fejlesztési lehetőségek

A fejlesztés során több ötlet is felmerült bennem, hogy milyen irányba lehetne továbbfejleszteni az alkalmazást. Milyen plusz funkcionalitásokkal bővüljön a szoftver és a meglévő funkcionalitásokat hogyan lehetne megváltoztatni annak érdekében, hogy minél szélesebb igényeket szolgáljon ki a szoftver.

Néhány ötlet új funkciók tekintetében:

- Saját receptek felvétele, törlése, módosítása.
- Bevásárlólista aloldal készítése, ahol a felhasználó kiválaszthatja és kinyomtathatja a naptárban szereplő ételek receptjeit.
- Email-ben történő értesítés a felhasználó számára, annak érdekében, hogy milyen napi receptek következnek majd.
- Ételallergia és nem kívánt ételcsoportok felvétele és kezelése egy – egy adott felhasználó esetében.
- A statisztikák menüpont alatt nem csak múltbéli, hanem megjósolt adatok megjelenítése a felhasználó számára.

Módosítást igénylő funkciók:

- A saját adatok menüpont alatt, a már kitöltött mezők módosításának engedélyezése.
- Az étrendek ütemezése menüpontban a kiválasztott étrend a felhasználó által kijelölt napra történő elhelyezésének biztosítása.
- Az naptárban lévő étrendek törlésének biztosítása.
- Az étrend tervezés során saját profil készítése, annak érdekében, hogy ne legyen szükség minden belépésnél kitölteni a kérdéseket.
- Az étrendjeim menüpont alatt található étrendekben lévő receptek cserélhetősége.
- A receptek menüpontban lévő elemekre történő szűrés recept neve, ételmiszer típusa vagy ételmiszer alapján.

11 Összefoglaló

Szakdolgozatom megírásával egy olyan, számomra fontos témát választottam, amely hosszú évek óta foglalkoztatott, ez az egészséges táplálkozás és életmód kialakítása. Célom az volt, hogy felépítsek egy olyan weboldalt, amely segítségével lehet azok számára, akik szeretnének odafigyelni táplálkozásukra.

Kiinduló pontként megvizsgáltam, hogy léteznek-e már ehhez hasonló rendszerek, s a talált megoldásokat elemeztem és összehasonlítottam egymással, illetve a saját ötleteimmel. Ez nagyban hozzájárult ahhoz, hogy megalapozzam a saját alkalmazásom funkcionalitásait.

A webalkalmazásom működéséhez elengedhetetlen volt, hogy bizonyos hiteles adatforrásokat alkalmazzak. Fontos volt a statisztikai adatok végett, hogy a receptek élelmi anyagainak tápértékei biztos forrásból származzanak.

Megvizsgáltam a lehetséges adatbázis típusokat annak érdekében, hogyan tároljam el az adatokat, úgy, hogy minél gyorsabban elérhetőek legyenek, ehhez összehasonlítottam az egyes lehetőségeket.

A specifikációban meghatároztam a céloknak megfelelően a webalkalmazás felhasználói követelményeit, ennek megfelelően felépítettem az alkalmazás funkcióit, melyeket részletesen kifejtettem. Elemeztem a saját fejlesztésű weboldalam előnyeit és hátrányait.

A céloknak megfelelően és a kutatásaim alapján határoztam meg, hogy milyen platformon jelenjen meg az alkalmazásom és ehhez milyen technológiai háttérrel biztosítsak. Választásom a PostgreSQL relációs adatbázis kezelő rendszerére esett, a szerveroldali szolgáltatásként a JAVA alapú Spring Boot keretrendszert, és a felület megjelenítéséért felelős Angular-t helyeztem előtérbe.

A specifikáció alapján készítettem el a rendszertervet, ami alapján felépül az adatbázis modell. A használati eset diagram segítségével bemutattam a felhasználói interakciókat és a rendszerterv modelljét. Végezetül bemutatásra került, hogy milyen képleteket alkalmaztam a kalkulációk megvalósításához.

A megvalósítás során bemutattam a szerver és a kliens oldali felépítést, illetve kifejtésre került, hogyan oldottam meg a felhasználói hitelesítést, és hogyan futtattam a Docker konténerben az egyes rendszereket. A szakdolgozatom utolsó fejezeteiben szemléltetésre került a rendszer működése, elvégeztem az End to End tesztelést és a szerveroldalon lévő néhány osztály unit tesztelését.

Megállapítottam, hogy a szoftver a specifikációnak megfelelt, és felvázoltam a továbbfejlesztési lehetőségeket, amelyeknek megfelelően kifejtettem az új ötleteimet és a meglévő funkciók módosításainak opcióit.

12 Summary

By writing my dissertation I have chosen a topic that is important to me, which I have been dealing with for a long time. This is a healthy diet and lifestyle developing. My goal was to build a website that could help those who want to pay attention to their nutrition.

As a starting point, I examined whether similar systems already exist, I analyzed and compared the found solutions and my ideas. This has greatly helped me establish the functionality of my own application.

It was essential for my web application to work with certain authentic data sources. It was important for the statistics that the nutritional value of the nutrients in the recipes came from a reliable source.

I analyse the possible database types in order to store the data so that it would be available as quickly as possible, for which I compared the individual options. I defined the user requirements of the web application in the specification according to the goals, I built the functions of the application accordingly, which I explained in detail. I analyzed the pros and cons of my self-developed website.

I determined which platform my application should appear on and what technological background to provide according to the goals and based on my research. I chose the PostgreSQL relational database management system, and I focused on the JAVA-based Spring Boot framework as the server-side service, and Angular, which was responsible for displaying the interface.

According to the specification I created the system design based on which the database model is built. Using the use case diagram, user interactions and the system design model were presented. Finally I presented the formulas which I applied to implement the calculations.

During the implementation I presented the server and client side structure, how I solved user authentication and how I ran each system in the Docker container. The operation of the system was illustrated in the last chapters of my dissertation, furthermore I finished the End to End testing, and testing some of the class on the server side with unit tests. I found that the software met the specification and outlined the upgrade settings to explain my new ideas and options for modifying existing features.

13 Irodalomjegyzék

[1] Dr. David Perlmutter Agyépítők - Hogyan gyógyítja és védi agyunkat az egészséges bélflóra - egy életen át. Kossuth Kiadó Zrt., 2017

[2] Dave Asprey Golyóálló diéta. Jaffa kiadó és kereskedelmi KFT., 2017

[3] Dr. David Perlmutter Agyépítők - Hogyan gyógyítja és védi agyunkat az egészséges bélflóra - egy életen át. Kossuth Kiadó Zrt., 2017

[4] Dave Asprey Golyóálló diéta. Jaffa kiadó és kereskedelmi KFT., 2017

[5] Dr. David Perlmutter Agyépítők - Hogyan gyógyítja és védi agyunkat az egészséges bélflóra - egy életen át. Kossuth Kiadó Zrt., 2017

[6] Dave Asprey Golyóálló diéta. Jaffa kiadó és kereskedelmi KFT., 2017

[7] NutriComp
<https://www.nutricomp.hu>
Utoljára megtekintve: 05.03

[8] Palatinus fitness Klub
<https://palatinustunde.hu>
Utoljára megtekintve: 05.03

[9] Way of Life
<http://www.wayoflife.hu>
Utoljára megtekintve: 05.03

[10] Okostányér
<http://www.okostanyer.hu/>
Utoljára megtekintve: 05.03

[11] Egészségfejlesztési Iroda | Sarkad Dokumentumok
<http://szakrendelo-sarkad.hu/efi/letoltheto-dokumentumok>
Utoljára megtekintve: 05.03

[12] Élelmiszerlánc-felügyeletért felelős Államtitkárság – Tápértékjelölésről – tápérték adatbázis ajánlások
<https://elelmiszerlanc.kormany.hu/tapertekjelolesrol-tapertek-adatbazis-ajanlasok>
Utoljára megtekintve: 05.03

[13] Holland RIVM weboldalon található egy ingyenesen letölthető táblázat
<https://www.rivm.nl/en/dutch-food-composition-database/nevo-online-request-dataset>
Utoljára megtekintve: 05.03

[14] Nemzeti Agrárgazdasági Kamara
<https://www.nak.hu/tajekoztatasi-szolgalattas/elelmiszer-feldolgozas/93268-tapertekadatbazisok-es-bovitesuk>
Utoljára megtekintve: 05.03

[15] Campden BRI Magyarország
<https://campdenbri.hu/kiadvanyok/tapertek-adatbazis.php>
Utoljára megtekintve: 05.03

[16] Relációs adatbázisok Oracle
<https://docs.oracle.com/en/database/>
Utoljára megtekintve: 05.10

[17] Kulcs-érték tárolók
<https://redis.io/documentation>
Utoljára megtekintve: 05.10

[18] Oszlopalapú tároló
<https://hbase.apache.org/>
Utoljára megtekintve: 05.10

[19] Dokumentumtárolók
<https://www.mongodb.com/>
Utoljára megtekintve: 05.10

[20] Gráfadatbázis
<https://neo4j.com/>
Utoljára megtekintve: 05.10

[21] PostgreSQL
<https://www.postgresql.org/docs/current/>
Utoljára megtekintve: 05.10

[22] Oracle JAVA Download
<https://www.oracle.com/java/technologies/javase-downloads.html>
Utoljára megtekintve: 05.10

[23] PostgreSQL

<https://docs.microsoft.com/hu-hu/azure/postgresql/>

Utoljára megtekintve: 05.10

[24] Java Download information

<https://www.java.com/en/download/manual.jsp>

Utoljára megtekintve: 05.10

[25] Java Virtual Machine

<https://www.educba.com/java-virtual-machine/>

Utoljára megtekintve: 05.10

[26] Nagy Gusztáv Java programozás 1.3 verzió 2007. február

https://nagygusztav.hu/sites/default/files/csatol/java_programozas_1.3.pdf

Utoljára megtekintve: 05.10

[27] Spring Boot

<https://sanfranciscoboljottem.com/tanfolyamok.html>

Utoljára megtekintve: 05.10

[28] Spring Boot

<https://spring.io/projects/spring-boot>

Utoljára megtekintve: 05.10

[29] Angular

<https://angular.io/docs>

Utoljára megtekintve: 05.10

[30] STS

<https://spring.io/tools>

Utoljára megtekintve: 05.10

[31] VSC

<https://code.visualstudio.com/>

Utoljára megtekintve: 05.10

[32] WebBeteg

<https://www.webbeteg.hu/cikkek/egeszsegedre/19226/uj-hazai-taplalkozasi-ajanlas>

Utoljára megtekintve: 05.10

- [33] Okostányér - Rólunk
<https://www.okostanyer.hu/rolunk/>
Utoljára megtekintve: 05.10
- [34] Tápértékjelölés, Tápérték adatbázisok
<https://campdenbri.hu/kiadvanyok/tapertek-adatbazis.php>
Utoljára megtekintve: 05.10
- [35] Dr. Balázs Péter, Adatbázisok
http://eta.bibl.u-szeged.hu/3387/1/BalazsNemeth_Adatbazisok_jegyzet_2019.pdf
Utoljára megtekintve: 05.10
- [36] Testtömeg-index kiszámítása
<https://www.okostanyer.hu/energia-egyensuly/>
Utoljára megtekintve: 2022.01.17
- [37] Testzsír százalék kiszámítása
<https://www.icliniq.com/tool/body-fat-percentage-calculator>
Utoljára megtekintve: 2022.01.16
- [38] Minimális energiaigény
<https://gymbeam.hu/blog/hogy-szamoljuk-ki-az-alapanyagcsere-erteket-es-a-napi-energiafelhasznalast/>
Utoljára megtekintve: 2022.01.17
- [39] UUID generated
<https://www.uuidgenerator.net/>
Utoljára megtekintve: 2022.01.18
- [40] Nosalty
<https://www.nosalty.hu/mi-van-a-hutodben>
Utoljára megtekintve: 2022.01.18
- [41] Spring Boot dokumentáció
<https://docs.spring.io/spring-boot/docs/current/reference/html/index.html>
Utoljára megtekintve: 2022.01.19
- [42] PostgreSQL JDBC Driver Introduction
<https://jdbc.postgresql.org/documentation/head/intro.html>
Utoljára megtekintve: 2022.01.19

[43] Keycloak adapter dokumentáció

https://www.keycloak.org/docs/latest/securing_apps/

Utoljára megtekintve: 2022.01.19

[44] Keycloak Spring Boot-al

<https://www.baeldung.com/spring-boot-keycloak>

Utoljára megtekintve: 2022.01.19

[45] Docker Desktop

<https://www.docker.com/products/docker-desktop>

Utoljára megtekintve: 2022.01.19

[46] Docker

<https://docs.microsoft.com/hu-hu/azure/cognitive-services/containers/docker-compose-recipe>

Utoljára megtekintve: 2022.01.19

[47] Keycloak

<https://www.keycloak.org/>

Utoljára megtekintve: 2022.01.19

[48] Material Design

<https://material.io/>

Utoljára megtekintve: 2022.01.19

[49] Step-by-step guide how integrate Keycloak with Angular application

<https://wkrzywiec.medium.com/step-by-step-guide-how-integrate-keycloak-with-angular-application-d96b05f7dfdd>

Utoljára megtekintve: 2022.01.19

[50] Keycloak - Angular

<https://www.npmjs.com/package/keycloak-angular>

Utoljára megtekintve: 2022.01.19

14 Ábrajegyzék

3.1. ábra: Mi legyen egy nap a tányérodon.....	12
3.2. ábra: Ajánlott ételtípusok.....	12
3.3. ábra: Élelmianyagaink kalória- és vitamin táblázata	13
6.1. ábra: JVM saját készítésű ábra.....	23
7.1. ábra: Étrendtervező recept EK modell tervezete	26
7.2. ábra: Étrendtervező felhasználó EK modell tervezet.....	27
7.3. ábra: Use Case Diagram	30
7.4. ábra: Rendszerterv modellezése.....	31
8.1. ábra: Alkalmazásom moduláris felépítése	34
8.2. ábra: Összeállított receptek az Excelben adatbázisba történő beimportáláshoz	36
8.3. ábra: Felhasználót érintő táblák	36
8.4. ábra: Recepteket érintő táblák.....	37
8.5. ábra: Étrendtervező rendszer moduláris felépítése	38
8.6. ábra: Alkalmazás rétegek elhelyezkedése.....	40
8.7. ábra: Étrend naptár API interface	43
8.8. ábra: Swagger webes felülete	43
8.9. ábra: Felhasználó hitelesítés modul	44
8.10. ábra: Adminisztrátor bejelentkezési felülete.....	45
8.11. ábra: Kliens beállítása.....	46
8.12. ábra: Szabály létrehozása.....	46
8.13. ábra: Általam létrehozott szabály alapértelmezetté állítás.....	47
8.14. ábra: Anuglar alkalmazás moduláris felépítése	48
8.15. ábra: A főoldal	50
8.16. ábra: Energiaigény kalkulátor.....	51
8.17. ábra: Elemek az első belépést követően	51
8.18. ábra: Már kitöltött antropometriai értékek által látható eredmények	52
8.19. ábra: Nem elérhető funkció.....	52
8.20. ábra: Saját adatok megadása	53
8.21. ábra: Kitöltött felhasználói adatok.....	53
8.22. ábra: Étrend tervezés folyamata.....	54
8.23. ábra: Generált nem mentett étrend.....	55
8.24. ábra: Elmentett étrendek	56
8.25. ábra: Étrend naptár.....	57
8.26. ábra: Részletes recept leírások.....	58
8.27. ábra: Vitaminok statisztikai adatai.....	60
8.28. ábra: Ásványi anyagok vas statisztikai adata.....	60

15 Táblázatok

1.1. Életminőség	3
2.1. Nutricomp összefoglaló	9
2.2. Palatinus Fitness klub összefoglaló	10
2.3. Way of Life összefoglaló	10
5.1. Saját fejlesztésű online felület előnyei és hátrányai	21
6.1. MySQL és PostgreSQL összehasonlítása	22