

Szegedi Tudományegyetem

Informatikai Intézet

Recipe hoarder webes alkalmazás

(Recipe hoarder web application)

Szakdolgozat

Készítette:

Vas Laura

gazdaságinformatika szakos
hallgató

Témavezető:

Dr. Bilicki Vilmos

egyetemi adjunktus

Szeged

2021

Tartalomjegyzék

Feladatkiírás	4
Tartalmi összefoglaló	5
Motiváció	6
1. Piacfelmérés	7
1.1. Grocy	7
1.2. Delish	7
1.3. Yummly	8
1.4. BigOven	8
1.5. ChefTap	8
1.6. Összefoglaló	9
2. Funkcionális specifikáció	10
2.1. Bejelentkezés/Regisztráció	11
2.2. Kezdőoldal	11
2.3. Recept saját gyűjteményekbe	12
2.4. Recept importálás	13
2.5. Recept kalória számlálás	14
2.6. Bevásárlólista	15
2.7. Bevásárlólista ajánló	16
3. Felhasznált technológiák	17
3.1. Angular	17
3.2. Angular Material	17
3.3. FireBase	17
3.4. PWA	18
3.5. Schema.org	18
3.6. Figma	19
4. A rendszer magas szintű áttekintése	20
4.1. Recept importálás	20
4.1.1. FetchUrlData	20
4.1.2. JsonLdExtractor	20
4.1.3. SeparateIngredients	20
4.1.4. ImageDownloader	20
4.1.5. CalorieCalculator	20
4.2. Kliens oldali adatbázis műveletek	20
4.3. Bevásárlólista ajánló	20

5. Architektúra	21
5.1. asd	21
6. Adatmodellek	22
6.1. Recipes	22
6.2. Users	23
6.3. Reports	24
7. Tesztelés	25
7.1. asd	25
8. Továbbfejlesztési lehetőségek	26
8.1. asd	26
Nyilatkozat	27
Köszönetnyilvánítás	28
Irodalomjegyzék	29

Feladatkiírás

A szakdolgozat során egy Angular keretrendszerben kialakított webes alkalmazás létrehozása volt a feladatom. A projekt a Firebase-t használja adatbázisként. A fejlesztés során a legfőbb cél a recept importálás más honlapokról volt. Az importálás második legfontosabb lépése az alapanyagok szétválogatása, hogy később a bevásárlólistába helyezésnél a megegyező anyagok összeadódjanak.

Tartalmi összefoglaló

- **téma megnevezése:**
A szakdolgozat céljául kitűzött témám egy Angular-ban írt web applikáció, ami recept megjelenítésre és importálásra használható.
- **feladat megfogalmazása:**
Az importálás funkció lehetővé teszi, hogy a " felhasználók egy helyen gyűjtsék a receptjeiket. Továbbá a receptek összetevőit egy bevásárló listába ki tudják menteni, ezzel is megkönnyítve a mindennapi életet. A felhasználók a többiek által létrehozott receptek között tudnak keresni, és a nekik tetsző recepteket ki " tudják menteni a saját receptgyűjteményeikbe.
- **megoldási mód:**
Az applikáció egy weblap formájában lett megvalósítva, mivel így lehet a legtöbb eszközt elérni egyetlen kód bázissal. A megvalósításhoz a már említett Angular keretrendszert használtam, illetve a Firebase felhő alapú szolgáltatásait. Mivel mind a kettő (Angular, Firebase) a Google terméke, ezért várhatóan hosszútávon támogatva lesznek. A felhasználó a recept URL-je alapján tud, receptet importálni, vagy manuálisan is tud létrehozni újat. Ekkor az importáláshoz egy szervert oldali funkció fut le és próbálja értelmezni a megkapott URL-en lévő html fájlt. Ennek egy fontos lépése az, hogy az alapanyagok nevét, mértékegységét és mennyiségét az eredeti kiolvassa. Ehhez regex-et illetve egy külső könyvtárat használtam, ami sok mértékegység között tud átváltani. Miután a receptet sikeresen importáltuk, azokat a Firebase Firestore adatbázisában tároljuk.
- **alkalmazott eszközök, módszerek:**
Mind az importálás mind az egész projekt során törekedtem, hogy minél modulárisabb legyen a felépítés. A webapp fejlesztése során a PWA-t alkalmazva elérhető, hogy " bizonyos funkciók offline is működjenek. A modern, könnyen kezelhető weblap számítógépen és telefonon egyaránt használható.
- **elért eredmény:**
A fejlesztés során sikerült egy modern telefonos és számítógépes környezetben is elérhető webes alkalmazást készíteni, ami bárki számára a regisztráció után elérhető és könnyen használható, ezzel könnyebbé téve a hétköznapiakat.
- **kulcsszavak:**
Angular, Firebase, pipeline architektúra, PWA, telefonos nézet

Motiváció

Egyetemisták, mint én is egyre közelebb vagyunk ahhoz az életformához, ahol önellátók vagyunk, ennek fontos része a főzés és étkezés. Manapság nagyon egyszerű különböző recepteket, különböző országokból, kultúrákból találni, viszont ez temérdeknyi weblapot jelenthet. Ennek hátulütője, hogy egy idő után követhetetlen lesz, hogy egyáltalán hova regisztráltunk, valamint, hogy “melyik weblapon is volt az a bizonyos recept, amit egyszer már kipróbáltam, és tetszett”. Személyes tapasztalatom ezzel kapcsolatba pedig, hogy én egy TXT fájlba mentettem az URL címeket, hogy legközelebb is megtaláljam, de már kezdett nagyon követhetetlen lenni.

Azért választottam ezt az ötletet a szakdolgozatom témájának, mert ez egy személyes problémám már hosszú ideje és láttam már korábban próbálkozásokat, de egyik sem volt az én elképzelésemnek megfelelő. A célom az volt, hogy egy egyszerű URL cím másolással pillanatok alatt egy helyen lehessen a megtalálni mindent.

A továbbiakban részletesen részletezem az általam tervezett és megvalósított webes applikáció felépítését és funkcióit. A bemutatót a konkurencia ismertetésével kezdem.

1. fejezet

Piacfelmérés

Már létező programokra öt példát hoztam, amik mind valamilyen szinten különböznek. Felhasználó körük, funkcióik, előnyök és hátrányok az én tervemhez képest.

1.1. Grocy

A Grocy egy lokálisan hostolható weblap. Irgalmatlanul részletes és rengeteg funkciója van, amihez, ha az ember hozzászokik és elég időt és törődést fektet bele, akkor egy nagyon hasznos program. Ellenben, mivel lokálisan van felépítve, ezért, ha valaki most kezdené el először használni, akkor nagyon sokáig tart, amíg igazán használható lehet.

A recept kezelő lapja csak manuálisan feltölthető, tehát nincs importálásra lehetőség. Rendelkezik bevásárlólista és “sufni” opciókkal is. Az otthon lévő alapanyagokat egyessével, tetsző részletességgel fel lehet venni a “sufniba”, ezzel leltározva, hogy milyen alapanyagok vannak otthon. Ezekről eltárolható adatok közé tartozik, hogy mennyi van belőle, meddig jók, képet, de akár a vonalkódját is. A bevásárló lista pedig egyértelműen a vásárlást segítő funkció, aminek a végén, egy kattintásra átrakható “sufniba”.

Már ezen kis leírás alapján is látszik, hogy ahhoz, hogy ez a rendszer használható legyen, egy komoly lokális adatbázist kell létrehozni az alapanyagokból és azok adatairól, valamint a receptekről. Ez a rendszer csak limitált tudású emberek számára használható, mivel már csak a telepítése is kicsit bonyolultabb, ezért átlag emberek számára nem ajánlott.

1.2. Delish

Ezt a weblapot azért választottam példaként, mert ez egy tökéletes példa egy átlagos, egyszerű receptes weblapokra. A honlapon csak recepteket és talán pár blog bejegyzés található regisztráció után is. Ez a weblap reprezentálja a legtöbb hasonló, csak blogként működőket.

Egy receptre kattintva látjuk az alapadatokat, hozzávalókat, elkészítési javaslatot valamint alap adatokat mint az elészítési idő. A weblaon található még hasonló recept ajánlások, de ezzel le lett fedve minden funkciója.

1.3. Yummly

Ez egy fejlettebb verziója a korábban említett "átlagos" weblapoknak. Bejelentkezés nélkül egy kissé korlátozott, viszont utána már kifejezetten sok képessége van. A webes kinézetén felül applikációval is rendelkezik.

Az alap recept keresésén kívül, itt már lehetőségünk van azok elmentésére a sajátjaink közé. A weblap rendelkezik bevásárló lista funkcióval, valamint képes azonnal a receptből áthelyezni az alapanyagokat is. Egy kiemelkedő funkciója az étkezés tervező. Ez, figyelembe véve esetleges allergiákat, vagy étrendeket ajánl és segít tervezni a következő időszakra.

Ami hátrány az egész weblapon, hogy nem közösség bővíti a recept adatbázist, ezért limitált a receptek száma és nem lehet mindent megtalálni. Még akkor is, ha figyelembe vesszük a manuális recept készítést, nem feltétlenül a legfelhasználóbarátibb, hogy mindig egy külső helyről egyösször kikeressük amit akarunk, majd kézzel beírjuk.

1.4. BigOven

A legnagyobb különbség az eddigiekhez képes, hogy ez a weblap már rendelkezik recept importáló funkcióval is. Azon felül négy különböző módon lehet újjakat létrehozni. Az importálás során nem tárolják el az egész receptet, ha más honlapról származik. "Our Pledge to Food Bloggers" leírja, hogy miért, viszont ez azt jelenti, hogy a teljes receptet megtekintsük, át kell navigálni az eredeti oldalra. Ezen felül a bevásárló listában nem adódnak össze a termékek, valamint nincsenek kategóriák a receptekhez.

Egy nagy hátránya a weblapnak, hogy kissé régi stílusú. A gombok majdnem eredeti HTML alakban jelennek meg. A képek, form-ok, lista nézet mind úgy néz ki, amin épphogy van egy kis formázás. A webes kinézetet a navigációja nem a legfelhasználóbarátibb.

Annak ellenére, hogy a weblapnak mennyire nem modern stílusa van, az applikáció igenis követhető. A funkciók szintén jól működnek. Elméletileg IOS-en is létezik, viszont arra nincs lehetőségem, hogy felmérjem milyen különbségek lehetnek. Egy nagy előny, hogy az ingyenes verzióban is használhatóak az alapfunkciók.

1.5. ChefTap

Az összes közül valószínűleg ez az applikáció, ami a legtöbb funkcióval rendelkezik. Technikailag van webes és telefonos applikációs verziója is, viszont a webes csak recept lekérdezésre használható. Minden egyéb, beleértve a recept importálást, bevásárló listát, étkezés tervezőt csak az applikációk keresztül lehet elérni és szerkeszteni. A weben volt lehetőség Google segítségével bejelentkezni, viszont az applikációnak nem volt ilyen lehetősége. Ennél az appnál az ingyenes verzió elég limitált, a recept importáláson kívül semmi sem működik a próbaidőszak lejártá után.

Ezen a felületen nincs mások által, vagy akár csak egy közös adatbázisból való keresésre és importálása lehetőség a recepteknél. A felhasználónak mindent magának kell beszerezni.

A beimportált recepteket könnyű módosítani, valamint rengeteg kis adatot megadni, hogy otthonosan lehessen használni a környezetet. Itt nincs lehetőség közvetlenül

a receptből a bevásárló listába rakni alapanyagokat, menüket összekészíteni vagy az étkezéstervezőt használni az ingyenes próbaverzió után.

1.6. Összefoglaló

Egy táblázat a korábban összefoglalt példákról és tudásaikról egy egyszerű összehasonlításhoz.

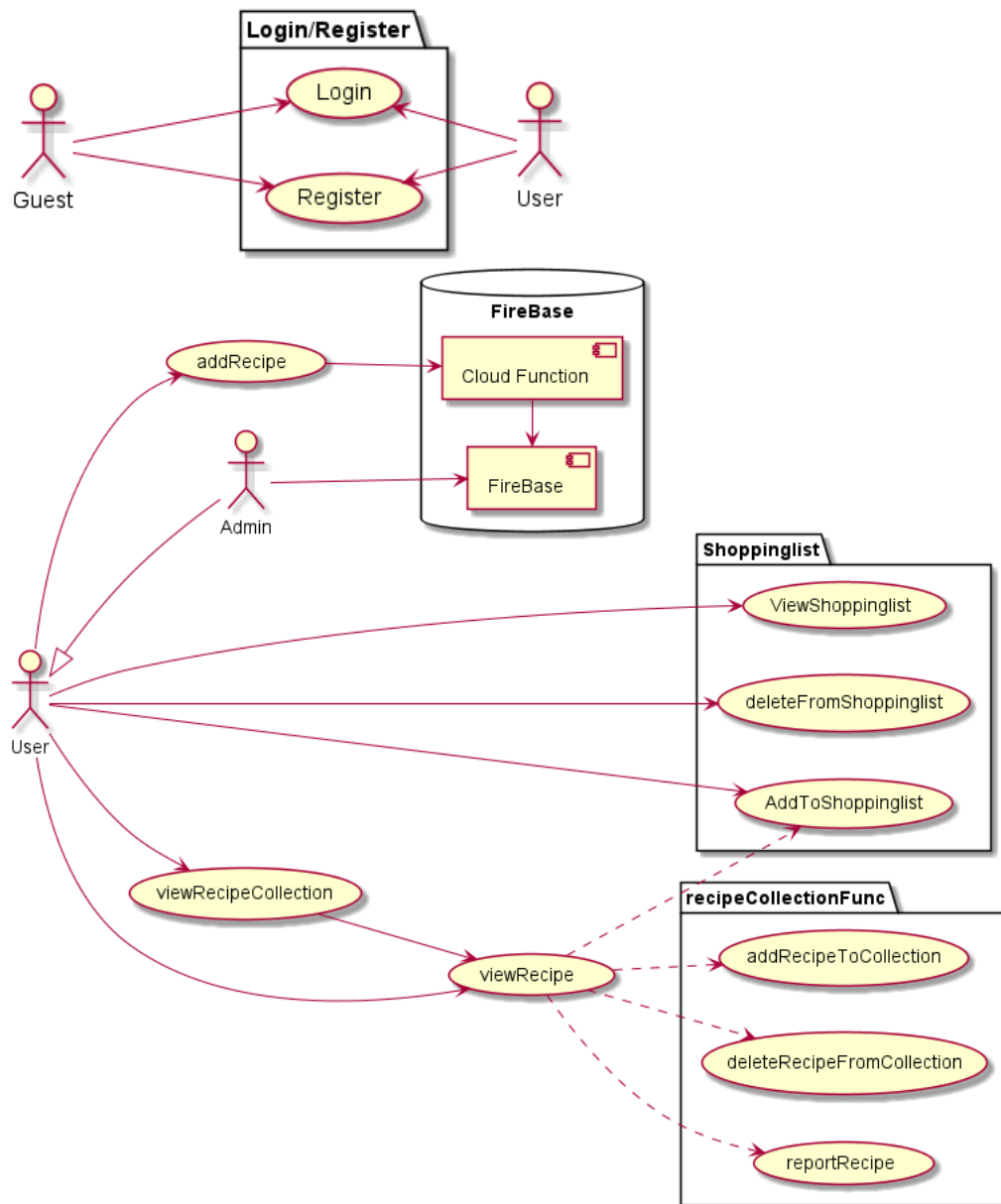
Összegzés

weblapok	Grocy	Delish	Yummly	BigOven	ChefTap	RecipeHoarder
open source	✓	✗	✗	✗	✗	✗
recept importálás - URL	✗	✗	✗	✓	✓	✓
receptekhez vannak lépések	✗	✓	✓	✗	✓	✓
manuális recept hozzáadás	✓	✗	✗	✓	✓	✓
bevásárlólista	✓	✗	✓	✓	✓	✓
bevásárlólista ajánló	✓	✗	✗	✗	✗	✓
kategóriák használata receptekhez	✗	✓	✓	✗	✓	✓
étkezés tervező	✓	✗	✓	✗	✗	✗
bejelentkezés google fiókkal	✗	✓	✓	✓	✗	✓
reszponzív web-lap	✓	✗	✗	✗	✗	✓

2. fejezet

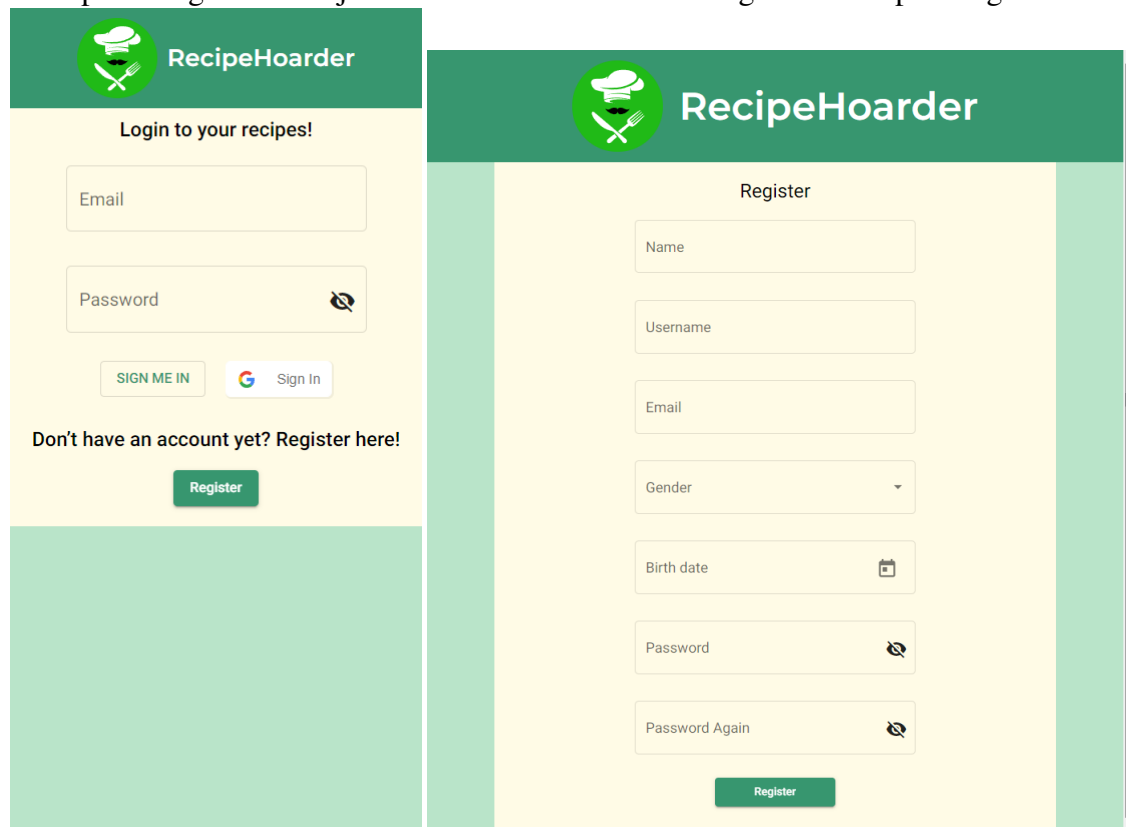
Funkcionális specifikáció

Egy összefoglaló részletesebb arról, hogy minek pontosan hogyan kell működnie az eredeti terv szerint.



2.1. Bejelentkezés/Regisztráció

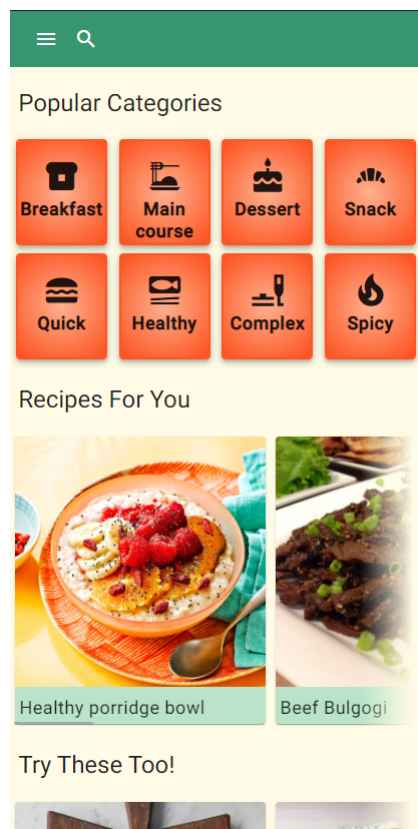
A regisztrációt és bejelentkezés a legbiztosabb biztonság érdekében a Firebase Auth rendszerén keresztül történik. Kettő módszer van a regisztrációra. Első, a szokásos email és jelszó páros megadásával a regisztrációs formon keresztül. Második, a Google autentikációs rendszeren keresztül. A felhasználók ezt látják először, mikor a honlapra navigálnak. Bejelentkezés nélkül nem lehetséges a weblapot megtekinteni.



The image displays two side-by-side screenshots of the RecipeHoarder web application's authentication interface. Both screens feature a green header with the 'RecipeHoarder' logo and name. The left screenshot shows the login page with the heading 'Login to your recipes!'. It contains input fields for 'Email' and 'Password' (with a toggle for visibility), a 'SIGN ME IN' button, a 'Sign In' button with a Google logo, and a 'Register here!' link with a 'Register' button below it. The right screenshot shows the registration page with the heading 'Register'. It includes input fields for 'Name', 'Username', 'Email', 'Gender' (a dropdown menu), 'Birth date' (with a calendar icon), 'Password' (with a toggle), and 'Password Again' (with a toggle). A 'Register' button is at the bottom.

2.2. Kezdőoldal

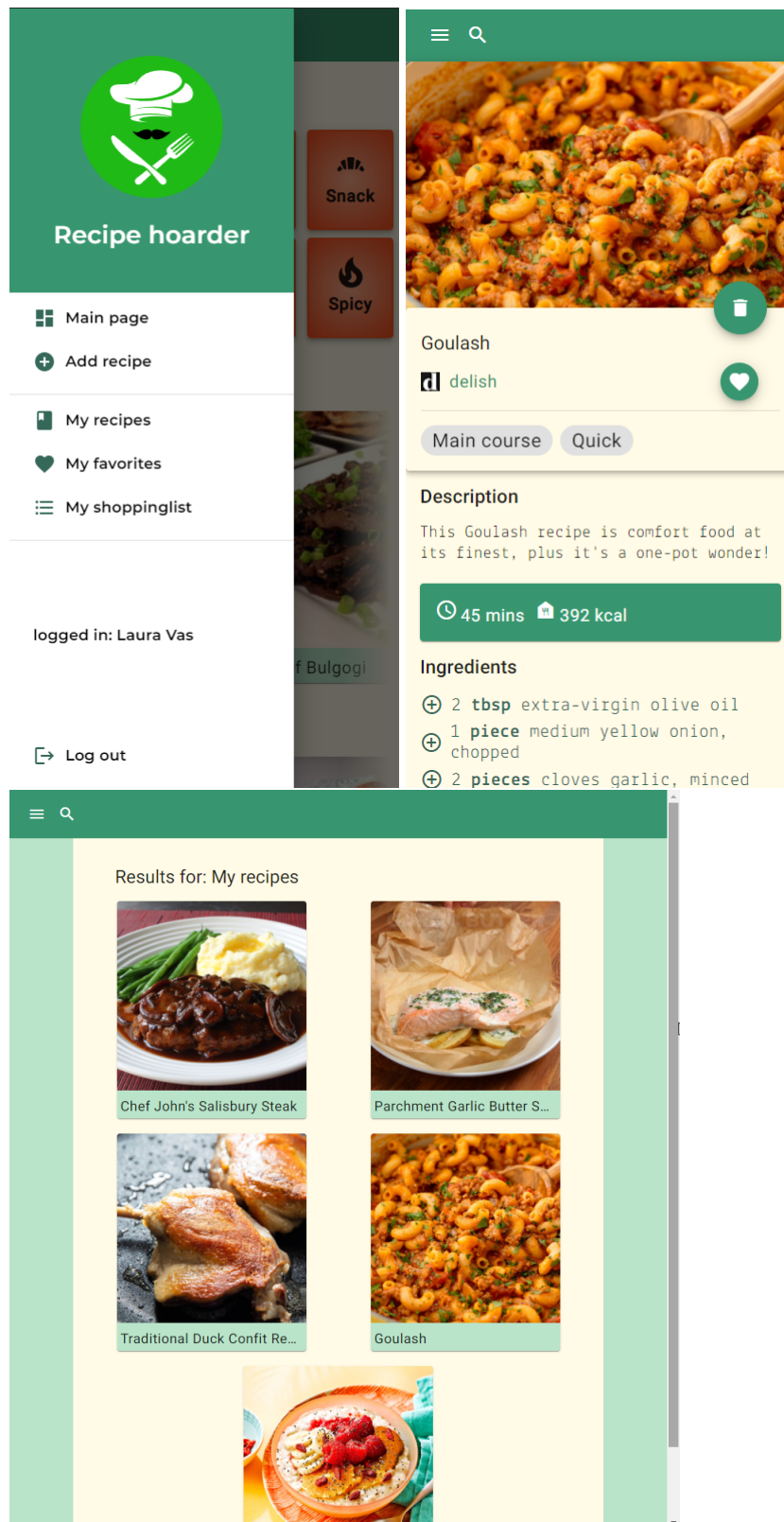
A kezdőoldalnak az első lap, amit a bejelentkezés után látnak a felhasználók. A lapon két különböző recept ajánló jelenik meg. Az egyik a saját, kimentett receptekből ajánlott fel párat, a másik viszont a még nem kimentett receptekből. Ezen felül a kategória keresés is itt érhető el. Minden receptnek létrehozáskor kötelezően van legalább egy kategóriája, ezért ez a típusú keresés jó az általános recept felfedező felhasználóknak.



2.3. Recept saját gyűjteményekbe

A létező recepteknél a megnyitás után egy gombnyomással lehetőségünk van azt a saját gyűjteményünkhöz adni. Amennyiben ez sikerült, a gomb átváltozik egy kuka ikonra. Ez újonnan megnyomása után a recept kikerül az elmentettjeink közül. Ez a lista egyszerűen megtalálható az oldal menüben a "My recipes" alatt. Ezen lap alatt az összes korábban elmentett recept kilistázódik.

Amennyiben a felhasználó kimentett egy receptet, a törlés gombra változásán kívül egy szívecske is megjelenik a recepten. Erre nyomva a recept egy külön listába kerül, ahol a saját receptek közül csak a kedvenceket lehet gyorsan eltárolni. A szívecskeire kattintással ki és be lehet kapcsolni, hogy a kedvencek közé kerüljön. Amint a felhasználó ki veszi a receptet a saját gyűjteményéből, akkor a kedvencek közül is eltűnik. A lista megtalálható az oldalsó menüben a "My favourites" menüpont alatt.



2.4. Recept importálás

Két féle képpen lehet recepteket létrehozni. Az elsőnél a felhasználónak csak ki kell tölteni a mezőket az új recept létrehozó felületen. Az alapanyagokból és lépésekből tetsző mennyiséget tud létrehozni, valamit törölni is.

A második módszer, ami a korábbi folyamat megkönnyítése miatt jött létre, a recept importálás URL-en keresztül. A felhasználó csak kiválaszt egy receptet egy másik honlapról, kimásolva az URL címét és az importálás folyamat során egy Cloud Function lekérdezi az adatokat. A funkció modulárisan lett felépítve, ezért jobban követhető, valamint bővíthető. Az importálás fontos része az alapanyagok szétválasztása alapanyag mennyiség, mértékegység és név szerint. Ez a receptnézetnél is megjelenik, de a fő célja a kalóriaszámláláshoz és a bevásárló listába helyezésnél kerül elő. A lekérdezés során, az egyik modul feladata a képek mentése az eredeti helyről, valamint feltöltése a Firebase Storage-ba. Ez fontos lépés, mert ha a weblap vagy recept egy ponton megszűnne, akkor is megmarad a kép és adatok a receptről. Amint a funkció lefutott, egy új lapra át lesz navigálva a felhasználó, ahol a recept szerkeszthető mezőkben megjelenik, ilyenkor még nincs elmentve az új adat az adatbázisba. Az importálás sikerességének átvizsgálása után, a felhasználó nyomhat a véglegesítő gombra.

A manuális és URL-es importálás ugyanazon a lapon működik, az egyetlen különbség, hogy üresen jut-e oda, vagy már lehetőleg egy sikeres lekérdezés után egy feltöltött verzióra. A véglegesítés előtt viszont, ha a felhasználó még nem választott besorolási kategóriát, akkor egy felugró ablak figyelmezteti, hogy ezt mindenképp tegye meg, mert nem léphet tovább anélkül.

2.5. Recept kalória számlálás

Ez a lépés egy modul a recept importálás folyamatban. A projekthez hozzáadtam egy csomagot, ahonnan egyrészt a szétválasztáshoz megkapom a mértékegységeket, másrészt ezt használok a mértékegység átváltáshoz is.

A tápanyag információkat egy külső adatbázisból szedem, ami az amerikai állami

Food Data Central API-ból elérhető. A használati feltételeket végigolvasva kiderült, hogy bárki ingyen használhatja, akár a jogok feltüntetése nélkül is, bár megköszönik, amennyiben ez feltüntetve lesz. Csak regisztrálni kell a weblapon ahhoz, hogy saját API kulcsot kapjunk.

A kalóriaszámlálás kizárólag akkor fut le, ha a receptlekérdezés során nem létezett az eredeti oldalon a kalória információ, vagy ha a recept manuálisan lett létrehozva. Ahhoz, hogy megfelelő eredményt kapjunk legalább 80 százaléknyi kiszámolt kalória adatok kell visszakapni az eredeti alapanyag mennyiségéhez képest. Ha ez nincs meg, akkor nem lesz megjelenítve kalória információ a recepthoz. Amennyiben megfelelő mennyiségű adatunk van, a korábban említett mértékegység átváltásokkal egységesítem az alapanyagokat a tápanyag táblázat értékéhez, aztán azt összeadva kapjuk a kalória adatot a recepthoz.

2.6. Bevásárlólista

A bevásárlólista egy külön lapon található, ahol a felhasználó a saját vásárlásához gyűjthet alapanyagokat. Ha a lapon található új alapanyag hozzáadó felületet használja, akkor hozzáadás után azonnal megjelenik a leni listában az új elem. Ha listában még létezett egy ponosan olyan nevű elem, akkor, ha szükség van rá, egy mértékegység átváltással hozzáadódik az eredeti elemhez. Így az alapanyag mennyisége nő. Amennyiben nem kell már az alapanyag, a sor végén lévő pipával ki lehet venni a listából.

A felhasználó kényelme érdekében, a receptből közvetlenül az alapanyag előtti plusz gombra nyomva hozzá lehet adni egyesével az alapanyagokat a bevásárló listába. Valamint ha mindent gyosan hozzá szeretnénk adni, akkor van egy másik gomb az alapanyag lista alatt, amivel az összeset egy kattintással hozzá lehet adni a listához.

Amount	Unit	Name	
0			+
1		spaghetti, cooked noodles, for serving	✓
0.5	cup	shredded mozzarella cheese	✓
0.25	cup	fresh basil leaves, sliced	✓
2	tablespoon	dried oregano, plus more to taste ...	✓
2	tablespoon	tomato paste	✓
1		spaghetti, cooked noodles, for serving	✓
0.5	cup	chicken stock	✓
1	package	salt	✓
10	Kg	Flour	✓
2	pieces	apple	✓
2	pieces	apple	✓

Remove all from shopping list

2.7. Bevásárlólista ajánló

A bevásárló listát használó felhasználók a lista megnyitásakor kapnak egy felugró lista elemet, ami egy korábbi alapanyagot kínál, amennyiben az több mind egyszer lett a listához adva. A pipára nyomva, az alapanyag bekerül a hozzáadó mező név elemébe. Amint a mennyiséget és mértékegységet kitöltve hozzáadjuk, akkor egyrész hozzáadódik az új elem a listához, valamint a következő alkalom dátuma újra generálódik a korábbiakat átlagolva.

Amennyiben a felhasználót nem érdekli az ajánlat, az x-re nyomva az ajánlat eltűnik, valamint elindul a funkció, ami -1 értékkel levonja az ajánlat értékelését. Az értékelés mező értéke alapvetően 2 mikor létrejön a meta adat. Amint az érték eléri a nullát, az alapanyag nem lesz többször felajánlva egy ideig.

Lehet, hogy ez már specifikáció inkább...

Annak érdekében, hogy ha a felhasználó egy ideig elvetette az alapanyagot, de egy idő után az mégis sokszor újra bekerül a listába, akkor minden hozzátételnél az értékelés mező +0.1 értékkel nő, ami lehetővé teszi, hogy egy idő után visszakerüljön az ajánlatok közé.

Az ajánlás egy sima adatbázis lekérdezés, ami szűrve van úgy, hogy csak ez a pillanat előtti "ajánlott időpont"-os alapanyagokat kérdezzünk le, ami rendezve van dátum és értékelés csökkenő sorrendjében. Végül a lekérdezés limitálva van három elemre.

The screenshot shows the 'Shopping List' app interface. At the top, there's a green header with a menu icon and a search icon. Below the header, the title 'Shopping List' is displayed in large red letters. Underneath, there are input fields for 'Amount' (with '0' entered) and 'Unit'. Below these is a 'Name' input field and a red button with a shopping cart icon. A section titled 'Suggestions:' follows, containing two rows: 'milk' and 'onion'. Each row has a green background, a checkmark button, and a red 'X' button. Below the suggestions, there are two rows showing the current list items: '6.18 l milk' and '1 dkg onion', each with a green checkmark button. At the bottom, there is a red button labeled 'Remove all from shopping list'.

3. fejezet

Felhasznált technológiák

3.1. Angular

Egy TypeScript alapú Framework fejlesztői platform, ami a eredetileg a Google fejlesztette ki, de most már Open Source. Angular egy teljes újraírása a régi AngularJS-nek. Egyik legnagyobb előnye, hogy komponens alapú a keretrendszer, ami segítségével újra felhasználhatók a kódok. Single Page application felépítésével dinamikusan építi fel a weblapok a felhasználó interakciója szerint, ezzel is gyorsabb és kevesebb újratöltést igényel.

A tervezés során a Figmában is komponensekkel terveztem meg a weblap kinézetét mind telefonos mind desktopos formában. A komponenseket egyszerű létrehozni, valamint felhasználni a megfelelő helyeken. Ha magunknak akarjuk tervezni a komponenseket, akkor sima HTML/CSS-el elérhető, amennyiben szeretnénk már egy népszerű és könnyen integrálható komponens bázist felhasználni, akkor az Angular Material használható.

3.2. Angular Material

Az Angular Material egy a Google által fejlesztett komponens bázis. Ez segít a fejlesztőknek már kész és modern dizájn komponenseket egyszerűen felhasználni a projekteken. Ez elsősorban egy UI komponens dizájn könyvtár, ami integrálás után, csak importokkal már azonnal működik. A weblapján pedig példákkal és részletes dokumentációval felsorolja a létező felhasználható komponenseket és különböző változataikat.

Az én projektem is nagy részben ennek a könyvtárnak a komponenseit felhasználva épült fel. Csak pár helyen lett kiegészítve a már létező verzió, hogy illeszkedjek az eredeti tervekhez. Ezen felül az Theming azaz a projekt témája is az Angular Material alapján lett felépítve, hogy a komponenseknek könnyen meglehessen adni az alap színeket és tipográfia szabályokat.

3.3. FireBase

Mostmár a Google által fejlesztett webes és mobilos alkalmazások készítéséhez létrehozott platform. Manapság nagyon népszerű az egyszerűsített adatbázis felépítése valamint a könnyű és könnyen értelmezhető kezelő felülete, arról nem beszélve, hogy rengeteg dokumentáció létezik a fejlesztők számára.

Ebben a projektben elég sok helyen használom a Firebase funkcióit. Az egész weblap a Firebase Hosting felületéről működik. A GitHub-bal összekötve, ha a Main Branch-re van feltöltés, akkor a Firebase-en automatikusan deploy-oldódik az új production verzió.

A Firestore Database-t használom a weblap adatok tárolásához. Itt vannak eltárolva receptek, felhasználók, felhasználókhöz különböző kollekciók és bevásárló lista meta adatok. A Cloud Firestore-on belül lehet beállítani az adatbázis Rule-okat. Ezt azt jelenti, hogy be lehet állítani, hogy különböző kollekciókat kik érhetik el, milyen tevékenységek vannak engedélyezve nekik, valamint, hogy milyen jogokkal milyen adatbázis műveleteket lehet csinálni. Amíg nem volt autentikáció beállítva és a development folyamat volt, akkor csak egy dátumot frissítettem, hogy meddig lehet az összes felhasználónak bármit csinálni. Viszont most már le van tiltva a receptek törése, valamint bejelentkezés nélküli lekérdezések is.

A Firebase Storage-ban tárolódnak a receptekből kimentett képek, amik az import során kerülnek az adatbázisba.

A legnagyobb biztonság érdekében a Firebase Auth-ot használom a weblap bejelentkezés és regisztrációs műveletekhez. Így lehetséges manuálisan regisztrálni majd bejelentkezni, viszont, akinek van Gmail fiókja annak lehetősége van egy lépésben azonnal regisztrálni és belépni annak segítségével.

A recept importálás és a bevásárlólista egy része a Firebase Cloud Function keresztül szerver oldalon fut. Ez nem csak olcsóbb, mivel szerver oldali lekérdezések ingyenesek a kliens oldalival szemben, hanem az importálás során felmerült problémát is megoldotta. Az importálásnál felmerült, hogy egy honlap nem érdekez le adatot egy másiktól kliens oldalon keresztül, valamint a CORS okozta nehézségek elkerülése érdekében is hasznos volt. Azon kívül, mikor a bevásárló listába új alapanyag kerül, akkor egy trigger lefut a szerver oldalon, ami segítségével állítjuk elő az ajánlóhoz a metaadatokat.

3.4. PWA

PWA azaz Progressive Web Application egy koncepció, amit követve lehetséges megbízható, alkalmazható és telepíthető webes applikációkat fejleszteni. Az alap szabályait követve a weblap platformspecifikus alkalmazássá alakítható. A PWA lehetővé teszi az alkalmazások használatát bárhol, bármikor, bármilyen eszközön.

Ez eseten az Angular-t kellett kiegészíteni egy kevés plusz JavaScript kóddal, hogy ha az eszköznek nincs internet elérése, akkor a script alapján a localStorage-ba elmentett, valamint cache adatokat használja az indításhoz. Ezzel elérhető, hogy valamilyen szinten használható a weblap offline környezetben is.

3.5. Schema.org

A schema.org egy weblap, ami azért készült, hogy a közösség a központosított sémákat használja különböző weblapokon, e-mailekben, modellekben, adatbázisok tervezésében vagy sok más helyen. Sok különböző kódolással használható, de a legnépszerűbbek a JSON-LD, Microdata valamint a RDFa. Ezek a szótárak olyan entitásokat és kapcsolatokat írnak le, amik segítik a fejlesztők számára a megfelelő sémát kiválasztani. Tehát összességében, ez egy tervekgyűjtemény.

A projektben fontos szerepet játszott a schema.org által biztosított “Recipe” séma feldolgozása és megértése, mivel a recept importálás más weblapokról ezen alapul. A jól felépített weblapok manapság tartalmaznak egy vagy több JSON-LD meta információt, amiben a weblapon jelenleg megjelenő adatok vannak eltárolva JSON formátumban a Schema.org szerinti felépítés alapján.

3.6. Figma

Egy vektorgrafikus szerkesztő, ahol web prototípusokat lehet tervezni. Ez legfőképp webalapú, de letölthető desktopos verziója is. A program képes a tervezett lapok élő tükrözésére akár saját telefonon az applikáción keresztül, vagy csak az emuláló felületen keresztül is. A figma egy modern dizájn tervező felület, ahova egyszerűen integrálni lehet a material design stílust és alap komponenseket, ezzel is segítve az Angular component szerű gondolkodást. Másik nagy előnye, hogy interaktívvá lehet tenni a tervezés során a gombokat, komponenseket. Valamint a kész dizájnoknak meglehet nézni a pontos CSS stílus kódját is, ezzel is segítve a konkrét kódolás folyamatát.

Sokat segített a tervezésben, mivel a kész material design komponenseket tudtam próbálgatni anélkül, hogy először mélyebben tanulmányoznám az Angular felépítését. Az egységes díjajn és téma felépítésében pedig segített konzisztensnek maradnom. Ez igaz mint a színekre, mind a tipográfiára.

4. fejezet

A rendszer magas szintű áttekintése

Ebben a fejezetben fogom leírni a különböző fontosabb funkciókról a részletesebb felépítését és tudnivalókat.

4.1. Recept importálás

A moduláris felépítés miatt ezt több al részre szedem.

4.1.1. FetchUrlData

asd

4.1.2. JsonLdExtractor

asd

4.1.3. SeparateIngredients

asd

4.1.4. ImageDownloader

asd

4.1.5. CalorieCalculator

asd

4.2. Kliens oldali adatbázis műveletek

Services...

4.3. Bevásárlólista ajánló

trigger és működés

5. fejezet

Architektúra

5.1. asd

6. fejezet

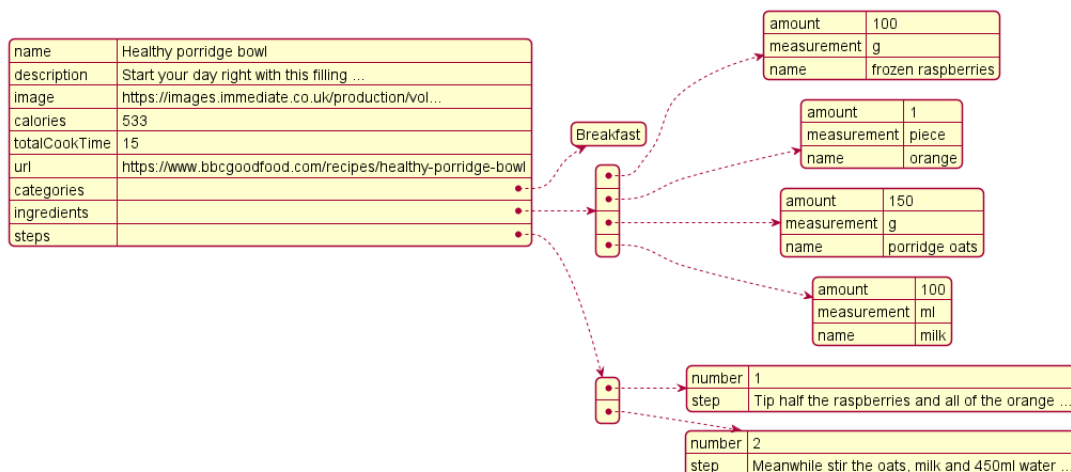
Adatmodellek

Az adatbázis felépítése különbözik a megszokott táblás módszertől, mivel a Firebase egy saját kifejlesztett módszerrel tárolja az adatokat, amivel gyorsabb a keresés és írás az adatok között. A szerkezete nagyon egyszerű, mivel csak egyszerű kollekciókból és dokumentumokból áll, felépítése hasonlít a JSON fájl típusokra. Minden adat egy-egy dokumentumban tárolódik, és ezek a dokumentumok vannak elrendezve kollekcióban. Ennek a felépítésnek a hátránya viszont, hogy nehéz az adatbázis struktúráját frissíteni, mert csak az új adatoknál fog megjelenni, ha nem indítunk rá egy külön funkciót. Ezért is macerásabb a régi adatokhoz változtatások után az új adattagba értéket adni.

6.1. Recipes

Az adatbázisban az egyik legfontosabb adat, mivel erre épül az egész weblap témája. A receptek a feltöltés során automatikus azonosító ID értéket kapnak, ami a weblapon az URL címében az azonosításért is felelős. Ezen belül tároljuk az adott recept adatait, ezek közé tartozik például: kalória, teljes főzési idő (percben mérve), recept neve, eredeti URL, stb. Ezek az alap adatok, csak simán kulcs-érték párok.

Jelenleg még kulcs-tömb értékben tárolva az alapanyagok és lépések, viszont a jövőben ez lehet megváltozik subcollection-re a gyorsabb betöltés idő eléréseért. Ezek a kulcs-tömb párosok hasonlóak a korábbi típushoz, viszont itt egy azonosító értékhez egy tömbbnyi belső adat tartozik. Ez lehetővé teszi, hogy az alapanyagokon belül eltárolhassuk egyesével a mennyiséget, mértékegységet és az alapanyag nevét. Valamint a recept elkészítésének lépésére el van tárolva a sorrendbeli szám és maga a lépés.



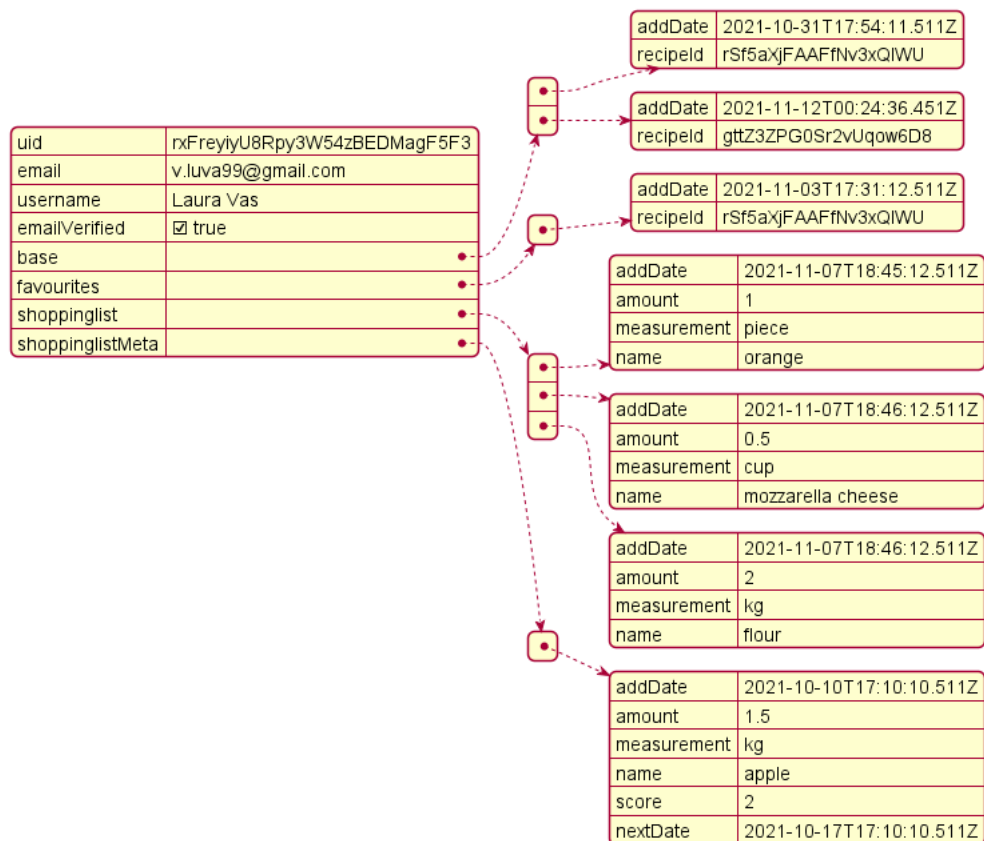
6.2. Users

Az adatbázisban a következő adathalmaz a felhasználók kollekciója. A regisztráció során létrejön egy új dokumentum az adatbázis felületen is. Itt tároljuk el az egyéb adatokat, amit a felhasználó az alap e-mail és jelszón kívül megad. Az egyéb adatok segíthetnek a jövőben személyre szabott recept ajánlót készíteni.

Itt a korábbi recepthez tartozó alapanyagok tömbhöz képest már egy subcollection-ban vannak eltárolva az egyéb gyűjtemények, amit a felhasználó a böngészés során elmentett. Ezek közé tartozik a kimentett receptek listája, kedvenc receptek, bevásárló lista valamint a bevásárló listához tartozó ajánlónak a metaadatai.

A subcollection azért fontos ebben az esetben, mert mikor le akarjuk kérdezni a felhasználó kedvenc receptjei listát, akkor nem kell mindig az egész felhasználó adatot lekérdezni az összes többi kollekcióval együtt, hanem lekérdezhető ugyanúgy, mint a többi kollekció egyszerűen. Ezzel gyorsabb lesz a weblap, valamint spórolunk a felesleges adatok kezelésével is. Amikor a felhasználó először elment a könyvtárába egy receptet, akkor a saját ID-ja alatt létrejön egy “base” kollekció, ami eltárolja, hogy mikor lett a listához adva a recept, valamint, hogy melyik lett kimentve a recept ID-ja alapján. Ugyanez igaz a “favourites” kollekcióra. Amint a felhasználó kimentett egy receptet a saját listájába, akkor van lehetősége ezt tovább elmenteni a kedvencek közé. Ez ugyanúgy működik mint a sima listába mentés. Amennyiben viszont a saját listájából törli a receptet, akkor a kedvencek közül is eltűnik.

A bevásárló listába a mentés dátumát és a pontos adatokat is elmentjük az alapanyag-ról, hogy utána az egész könnyen kilistázható legyen. A létrehozás dátuma a sorrendbe rendezés és a metaadat számítás miatt fontos. Ha a felhasználónak nem kell többé az adat, akkor törlődik a listából. Végül a metaadat, ami kettő listához adás értékéből számolunk ki. Az ajánló jelenleg csak az alapanyag nevét ajánlja fel, viszont a mértékegység és mennyiség is el van tárolva, hátha a jövőben pontosabb számítást szeretnénk csinálni. Ezen kívül, mikor az adat létrejön, a “score” értéke 2. Ez egy érték, ami segít az ajánlás sikerességét követni, max értéke 5, minimum értéke 0. Végül a “nextDate” egy számot TimeStamp érték, ami a eredeti hozzáadás dátumából és az újonnan hozzáadás dátumától számolunk ki.



6.3. Reports

Mivel a weblap úgy lett tervezve, hogy a közösség építi, és mindenki ugyanazokat a recepteket látja, ezért lennie kell egy lehetőségnek, ahol a felhasználók hibát tudnak jelezni. Tehát minden recept végén van egy felület, ahol jelzést lehet küldeni, ha hibás receptet találtak. A hiba bejelentés opciók előre be vannak állítva, így könnyebb lehet a jövőben megkeresni, hogy mi lehet a hiba az adott recepttel.

A “creationDate” a bejelentés elküldését jelzi. A “problem” pedig az opciókból választott a problémához legközelebbi lehetőség, amit a felhasználó választ. Valamint végül feljegyezzük, hogy melyik receptnél volt ez és hogy ki küldte. A “userId” meg egyezik a korábban említett “uid” értékkel a “users” kollekcióból, mivel ez az érték, ami a bejelentkezés során az auth adott a felhasználónak.

creationDate	2021-11-26T17:03:24.087Z
problem	Other
recipeld	1A3QX5z5n8nx2AWAo8ag
userId	rxFreyiyU8Rpy3W54zBEDMagF5F3

7. fejezet

Tesztelés

7.1. asd

8. fejezet

Továbbfejlesztési lehetőségek

8.1. asd

Nyilatkozat

Alulírott szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Tanszékén készítettem, diploma megszerzése érdekében.

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

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2021. november 26.

.....
aláírás

Alulírott szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Tanszékén készítettem, diploma megszerzése érdekében.

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

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a TVSZ 4. sz. mellékletében leírtak szerint kezelik.

Szeged, 2021. november 26.

.....
aláírás

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani **X. Y-nak** ezért és ezért . . .

Irodalomjegyzék

- [1] J. L. Gischer, The equational theory of pomsets. *Theoret. Comput. Sci.*, **61**(1988), 199–224.
- [2] J.-E. Pin, *Varieties of Formal Languages*, Plenum Publishing Corp., New York, 1986.