

## Projektmunka 2

GKLB\_INTM005

### Bevásárlólista alkalmazás

Csermák Cintia

XOEHP

Molnár Bertalan

N2A06U

Szandai Gábor Zsolt

A5R58R

2021/2022/1

## Contents

1.	Feladatkiírás .....	1
2.	Munkaeszközök .....	2
3.	Use Case-ek.....	2
4.	Funkciók részletezése.....	3
4.1.	Felhasználót érintő műveletek.....	3
4.2.	Listákat érintő funkciók .....	8
4.3.	Termék manipuláló funkciók .....	10
4.4.	Kimutatások.....	12
5.	Adatbázis felépítése.....	13
6.	Tesztek.....	16

# 1. Feladatkiírás

## Hallgatói csoport adatai: (3 fő)

Név	Neptun	E-mail	Szak	Tagozat
Csermák Cintia	XOEHZP	csermakcintia14@gmail.com	Mérnökinformatikus	Levelező/BSC
Molnár Bertalan	N2A06U	M.Bertalan@hotmail.com	Mérnökinformatikus	Levelező/BSC
Szandai Gábor Zsolt	A5R58R	szandai.zsolt@gmail.com	Mérnökinformatikus	Levelező/BSC

## Konzulens:

Hollósi János

## A projekt tárgyát képező probléma és a projektcél rövid felvázolása:

Bevásárló lista alkalmazás: egy olyan mobil eszközön futó alkalmazás elkészítése, amely alkalmas egy háztartás rendszeres bevásárlási listáit kezelni.

## Elvárt funkciók:

- Legyen lehetőség a listákra megvásárolandó termékeket felvinni, lehetőleg minél egyszerűbb módon.
- Célszerű, ha a lista több készülékről is elérhető, módosítható.
- Legyen lehetőség egy lista vagy annak egy részének teljesítettségét jelölni.
- Legyen meg a lehetőség, hogy az alkalmazás valamilyen módon előre jelezze, ha egy adott termékből hiány várható.
- Az alkalmazás készítsen kimutatást a háztartás vásárlási szokásairól, népszerű termékekről.

## Projektszervezet:

Az egyes feladatok csapaton belüli felosztását tartalmazza a következő táblázat:

	Név	Neptun	Feladatok
1	Csermák Cintia	XOEHZP	Adatbázisok
2	Molnár Bertalan	N2A06U	Tesztek, Dokumentáció
3	Szandai Gábor Zsolt	A5R58R	UI, Projektmenedzsment

A feladatfelosztás nem azt jelenti, hogy az adott személy egy maga a többiek segítségével nélkül kell elvégezze az adott feladatot, hanem azt, hogy az adott feladat koordinálása és a témához tartozó mélyebb kutatást a feladathoz megjelölt személy végzi.

## 2. Munkaeszközök

A projekt megvalósítására a Google Android Studio IDE-t választottuk, amely keretet biztosít az applikáció elkészítésének legfontosabb elemeihez. Tartalmaz kódszerkesztőt, továbbá grafikus felületet biztosít a UI gyorsabb megtervezéséhez. Beépített emulátoron keresztül tesztelési tevékenységekre is lehetőséget nyújt. A támogatott programnyelvek közül a java-t választottuk. A fejlesztői környezetet a csapat tagjai otthon saját számítógépükre ingyenesen telepíthetik és dolgozhatnak benne.

Az adatok tárolására a Google Cloud és a Firebase által üzemeltetett Cloud Firestore adatbázist használunk. Ennek SQLite-hoz viszonyított előnye például, hogy egy gyári szinkronizációs funkciót kínál. A szinkronizáció segítségével jóval egyszerűbben kivitelezhető, hogy több eszköz is egy közös online adatbázist használjon, ugyanakkor az applikáció offline is használható legyen. A szinkronizáción kívül gyári autentikációs funkciók is elérhetőek, amelyek segítenek a felhasználók azonosításában, a felhasználót érintő funkciók megvalósításban.

A projekt dokumentációit, az egyes modulokat és egyéb fájlokat, a verziókezelés érdekében GitHub repository-ban tároljuk.

## 3. Use Case-ek

A feladat előzetes elemzése alapján 19 használati esetet azonosítottunk, amelyek elkülöníthetőek a módosítani kívánt objektum alapján.

A felhasználói adatokat érintő funkciók többsége általánosan elvárható funkció, mint regisztráció, jelszókezelés, be-, illetve kijelentkezés. Ezen felül egy bevásárlólista közös kezelése érdekében egy egyszerű ismerős kezelő rendszert is létrehoztunk, ami segítségével a felhasználók egyszerűbben megoszthatják listáikat.

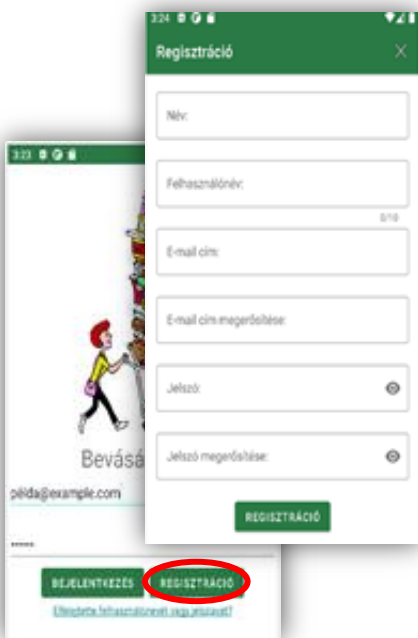
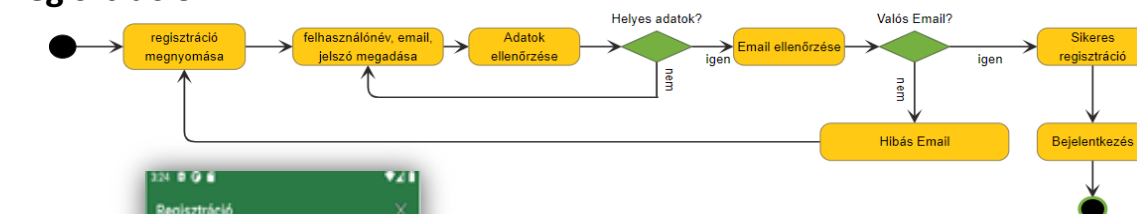
Természetesen a felhasználóknak kezelniük kell a listáikat is. Újat létrehozni, törölni, megosztani, nevüket módosítani. A bevásárlólistákon belül az egyes termékeket is tudni kell kezelni, felvenni, megvenni, törölni, módosítani. További követelmény a vásárlási szokásokról kimutatások készítése, és a statisztikák alapján bizonyos emlékeztetők, értesítők küldése.

Felhasználót érintő műveletek	Listákat érintő műveletek	Termékeket érintő funkciók	Kimutatások
<ul style="list-style-type: none"> <li>Regisztráció</li> <li>Bejelentkezés</li> <li>Felhasználó törlése</li> <li>Jelszó módosítása</li> <li>Új jelszó igénylése</li> <li>Ismerősök felvétele</li> <li>Ismerősnek jelölések fogadása/elutasítása</li> <li>Kijelentkezés</li> </ul>	<ul style="list-style-type: none"> <li>Lista létrehozása</li> <li>Lista nevének módosítása</li> <li>Lista megnyitása</li> <li>Lista megosztása</li> <li>Listamegosztás visszavonása</li> <li>Lista Törlése</li> </ul>	<ul style="list-style-type: none"> <li>Termék felvétele</li> <li>Termék törlése</li> <li>Termék adatainak módosítása</li> <li>Termék megvétele</li> </ul>	<ul style="list-style-type: none"> <li>Statisztikák megjelenítése</li> <li>Ajánlott termékek</li> </ul>

## 4. Funkciók részletezése

### 4.1. Felhasználót érintő műveletek

#### Regisztráció:



A felhasználót az alkalmazás megnyitáskor a bejelentkező képernyő fogadja (amennyiben még nem jelentkezett be.) Ezen a képernyőn lehetősége van új felhasználóként regisztrálnia magát.

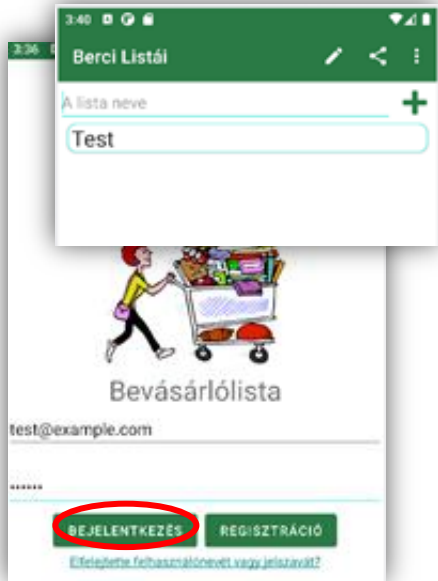
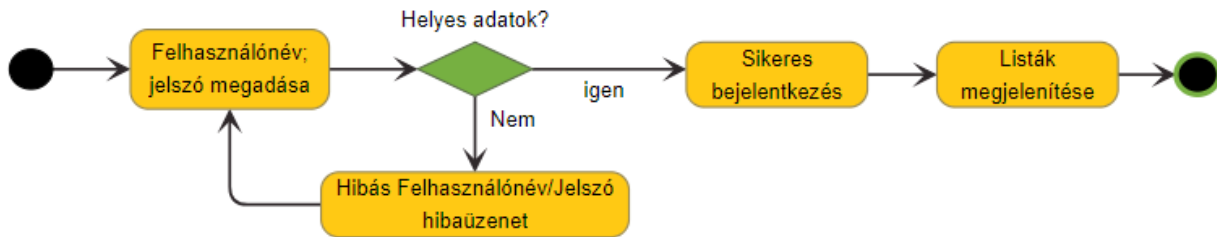
Regisztráció gomb megnyomásakor a regisztráló képernyő jelenik meg, ahol megadja a kívánt felhasználónevet, e-mail címet, jelszót.

Az adatok kitöltése után mentheti az adatokat. Ezt követően az UI réteg először ellenőrzi a bevitt adatok megfelelőségét, majd továbbítja az adatokat az adatbázis felé.

Az adatkezelő réteg új dokumentumot hoz létre az adatbázis felhasználók kollekciójában.

Az regisztrációs képernyő bezárásakor alkalmazás automatikusan bejelentkezteti az újonnan regisztrált felhasználót. Bejelentkezés után a Lista Választó képernyő jelenik meg.

## Bejelentkezés:



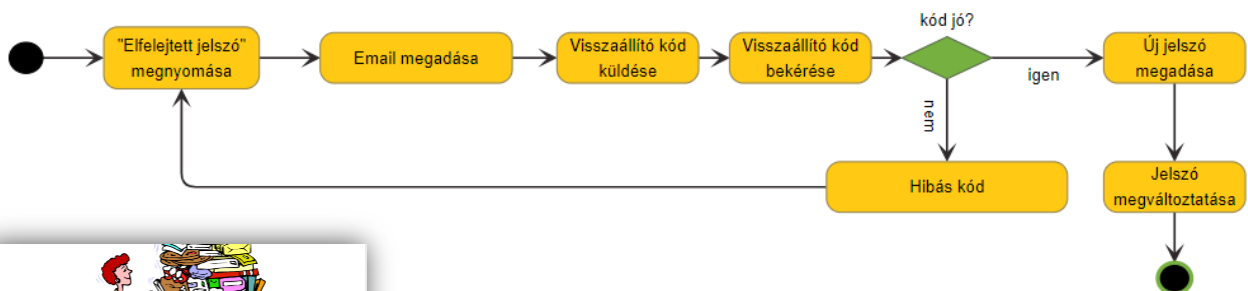
A felhasználót az alkalmazás megnyitáskor a bejelentkező képernyő fogadja (amennyiben még nem jelentkezett be korábban.) Ezen a képernyőn lehetősége van bejelentkezni.

A felhasználónév és jelszó adatok megadásával a felhasználó bejelentkezhet. A bejelentkezést követően a Lista Választó képernyő jelenik meg.

Bejelentkezéskor az applikáció továbbítja az adatokat az adatkezelő rétegnek, az megkeresi az felhasználó dokumentumát az adatbázisban.

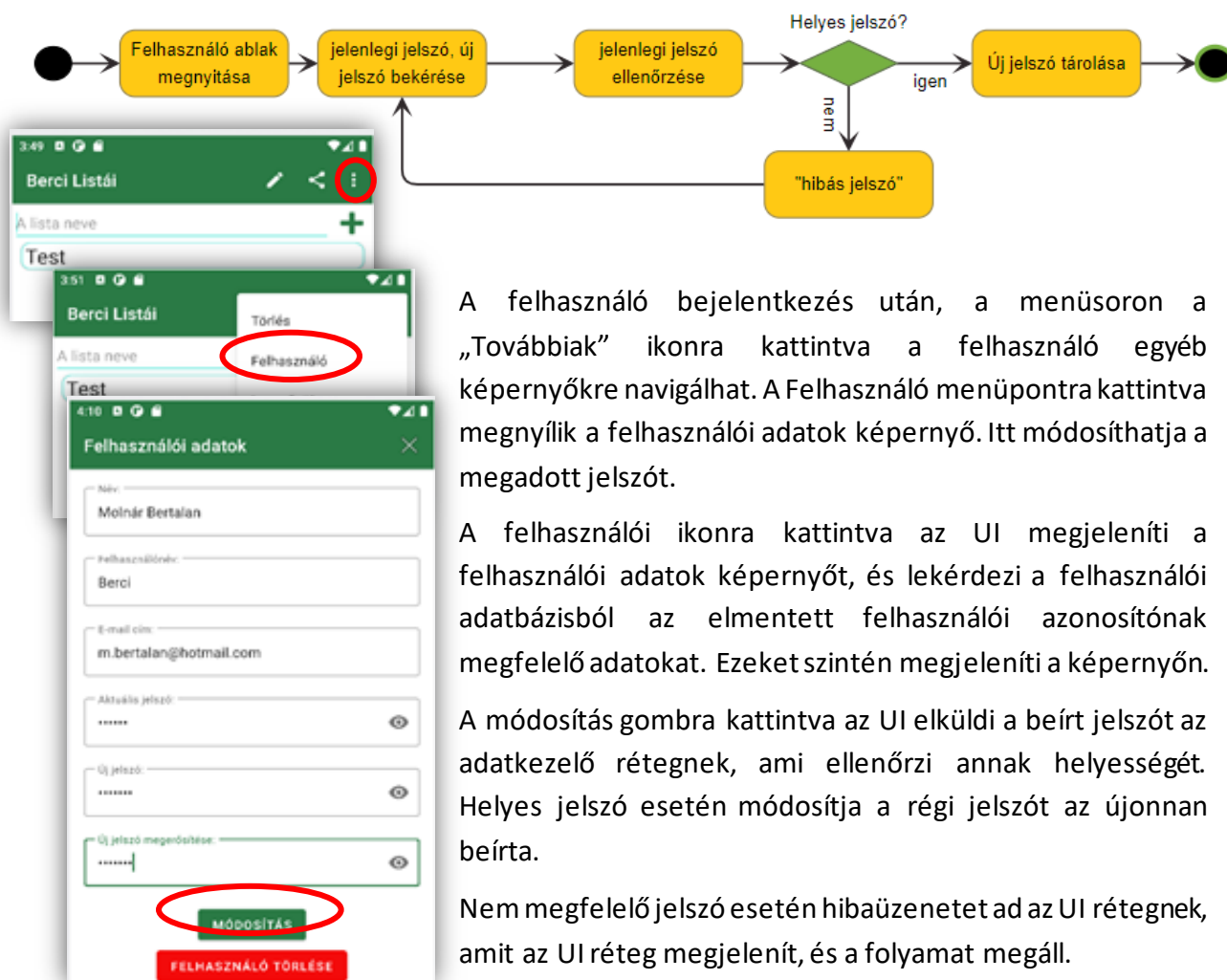
Hibás adatok esetén *Hibás felhasználónév/jelszó* üzenetet ad. Megfelelő jelszó esetén Az UI réteg megfelelő jelzést kap. Az applikáció elmenti a bejelentkezett felhasználó azonosítóját.

## Elfelejtett jelszó:



Bejelentkezés előtt a felhasználónak lehetősége van a jelszó helyreállítását kérni. Ezt az elfelejtett jelszó gomb megnyomásával teheti meg. Ezután meg kell adnia egy E-mail címet. Erre az E-mail címre egy automatikusan generált kód kerül kiküldésre, amit aztán az applikáció visszakérdez. Ha a felhasználó helyesen írja be a kódot, akkor a megadott e-mail címhez tartozó jelszót felülírhatja.

## Jelszó módosítása:



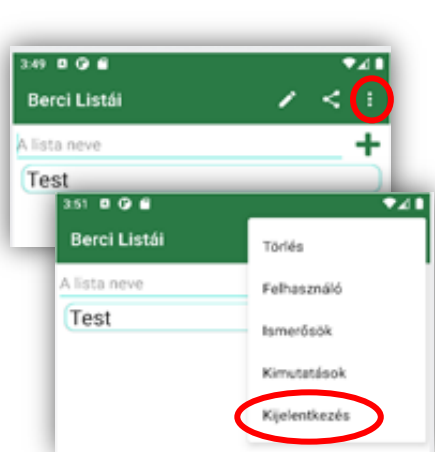
A felhasználó bejelentkezés után, a menüsoron a „Továbbiak” ikonra kattintva a felhasználó egyéb képernyőkre navigálhat. A Felhasználó menüpontra kattintva megnyílik a felhasználói adatok képernyő. Itt módosíthatja a megadott jelszót.

A felhasználói ikonra kattintva az UI megjeleníti a felhasználói adatok képernyőt, és lekérdezi a felhasználói adatbázisból az elmentett felhasználói azonosítónak megfelelő adatokat. Ezeket szintén megjeleníti a képernyőn.

A módosítás gombra kattintva az UI elküldi a beírt jelszót az adatkezelő rétegnek, ami ellenőrzi annak helyességét. Helyes jelszó esetén módosítja a régi jelszót az újonnan beírta.

Nem megfelelő jelszó esetén hibaüzenetet ad az UI rétegnek, amit az UI réteg megjelenít, és a folyamat megáll.

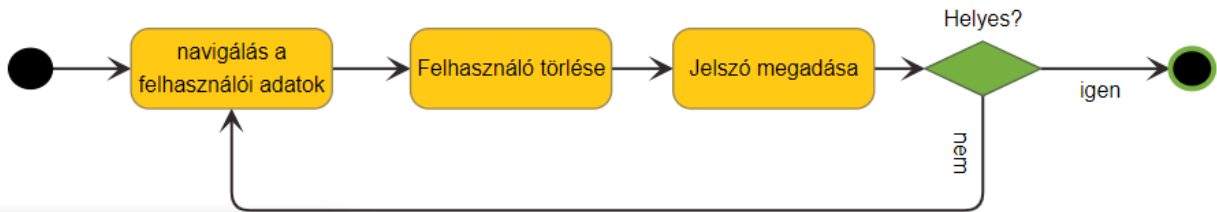
## Kijelentkezés:



A felhasználó bejelentkezés után a menüsoron a Felhasználói ikonra kattintva a felhasználó adatok képernyőre navigálhat. Itt lehetősége van kijelentkezni. A kijelentkezés gombra kattintva az alkalmazás kijelentkezteti a felhasználót és a bejelentkező képernyőt jeleníti meg.

A kijelentkezés gombra kattintva az UI meghívja az adatkezelő réteg kijelentkező funkcióját, ami törli az eltárolt felhasználói azonosítót.

## Felhasználói fiók törlése:



The screenshot shows a mobile app interface for user data management. It includes fields for Name (Molnár Bertalan), Username (Berci), Email (m.bertalan@hotmail.com), Current Password, New Password, and New Password Confirmation. At the bottom, there are two buttons: 'MÓDOSÍTÁS' (Modify) and 'FELHASZNÁLÓ TÖRLÉSE' (Delete User), with the latter circled in red.

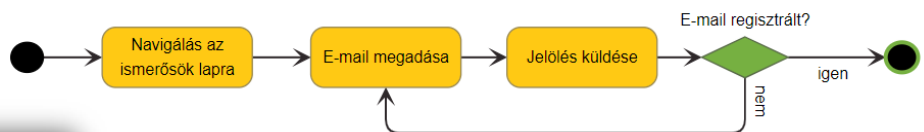
A felhasználó bejelentkezés után a menüsoron a Felhasználói ikonra kattintva a felhasználó adatok képernyőre navigálhat. Itt lehetősége van a felhasználói fiók törlésére. A törlés gombra kattintva a UI kéri a jelenlegi jelszót és megerősítést. Helyes jelszó, és megerősítés esetén kijelentkezteti a felhasználót, és törli a fiókot.

A fiók törlésére kattintva az UI megjeleníti a jelszó bekérő ablakot, majd elküldi a beírt jelszót az adatkezelő rétegnek, ami ellenőrzi annak helyességét. Helyes jelszó esetén *helyes jelszó* jelzést küld az UI rétegnek.

Nem megfelelő jelszó esetén *Hibás jelszó* hibaüzenetet ad az UI rétegnek, amit a UI réteg megjelenít, és a folyamat megáll.

Az UI ezután meghívja az adatkezelő réteg felhasználói fiók törlése funkcióját, ami kijelentkezteti a felhasználót, majd inaktíválja a felhasználói dokumentumot az adatbázis felhasználók kollekciójában.

## Ismerősök lap:



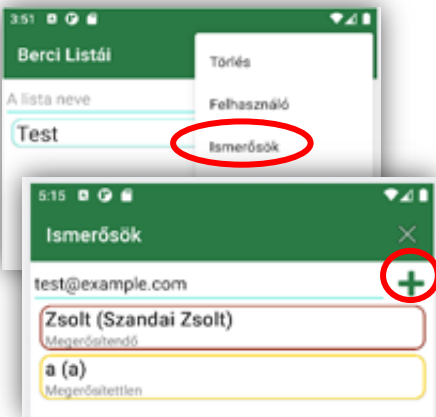
The first screenshot shows the 'Berci Listái' (Berci's Lists) screen with a list of lists. The 'Ismerősök' (Friends) list is circled in red. The second screenshot shows the 'Ismerősök' screen with a list of friends. The first friend, 'Zsolt (Szandai Zsolt)', is circled in red. Below the name, it says 'Megerősítendő' (Needs confirmation). The second friend, 'a (a)', is circled in yellow and says 'Megerősítetlen' (Not confirmed).

Az ismerősök lapra navigálva láthatjuk a visszaigazolt ismerőseinket (zöld), a még függő jelöléseinket (sárga), és a hozzánk beérkező felkéréseket (piros)

Lehetőségünk van új jelölések feladására, visszavonhatjuk a már elküldött jelölésünket, illetve mások jelölésének elfogadására/elutasítására. A visszaigazolt ismerőseinket is törölhetjük a listáról.



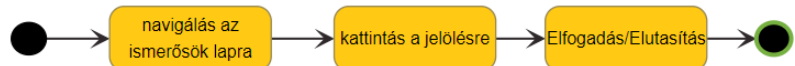
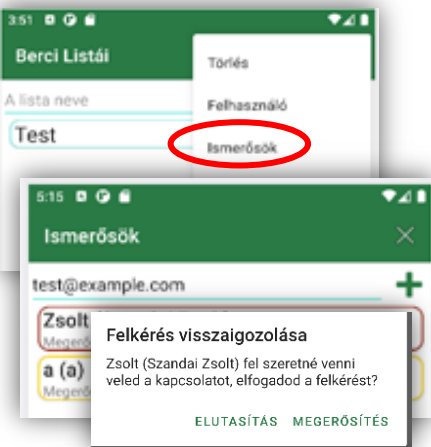
## Ismerősök bejelölése:



Az ismerősök lapon a kapcsolatunk e-mail címét beírva, majd a hozzáadás gombra kattintva elküldhetjük a jelölési szándékunkat.

Ekkor az UI elküldi az email címet az adatkezelőnek, ami kikeresi a kívánt felhasználó adatait. A Jelenlegi és a bejelölt felhasználó contactlist kollekciójában egy új dokumentum jön létre a megfelelő adatokkal. A jelenlegi felhasználó kollekciójában a ContactStatus mező értéke 1, míg a bejelölt felhasználó esetén a ContactStatus mezője 0 értéket vesz fel.

## Jelölések elfogadása/elutasítása:

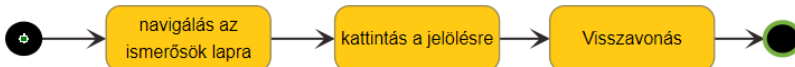
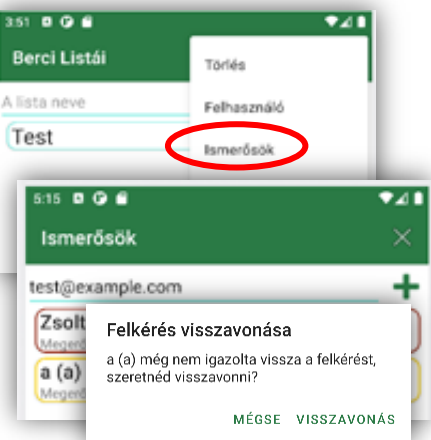


Az ismerősök lapon egy jelölésre kattintva lehetőségünk van elfogadni a jelölést.

Amennyiben így teszünk a két ContactList kollekcióban a ContactStatus értékek 2-re változnak.

Elutasítás esetén a két ContactList kollekcióból törlődnek a megfelelő rekordok.

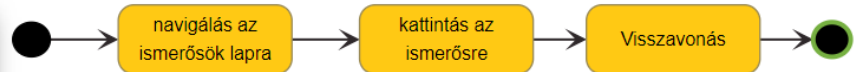
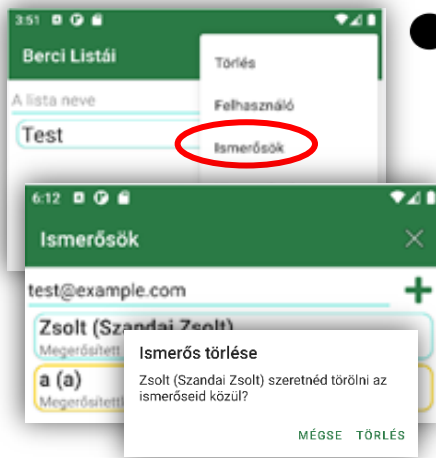
## Jelölés visszavonása:



Egy általunk feladott, még visszaigazolatlan jelölésre kattintva visszavonhatjuk a jelölést.

Ha így teszünk a megfelelő rekordok törlődnek a megfelelő felhasználók ContactList kollekcióból.

## Ismerősök törlése:

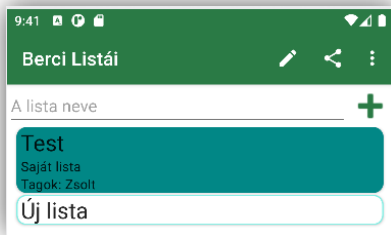


Egy ismerősünkre kattintva lehetőségünk van törölni az ismerősök listából.

Ha így teszünk a megfelelő rekordok törlődnek a megfelelő felhasználók ContactList kollekciójából.

## 4.2. Listákat érintő funkciók

### Lista Választó lap:

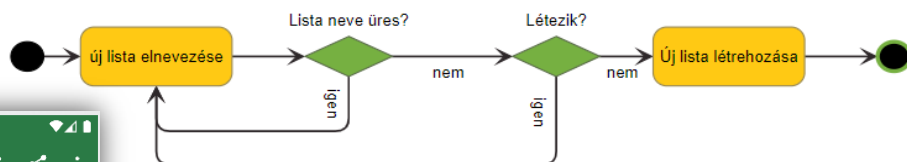
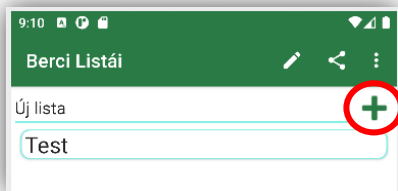


Bejelentkezés után, az applikáció megnyitásakor a lista választó lap nyílik meg. Ezen a felületen könnyen hozhatunk létre új listákat, illetve módosíthatjuk a már meglévőket.

A lapon egymás alatt láthatjuk a saját, és általunk hozzáférhető listákat. Megosztott listák esetén azok tulajdonosát, és Tagjait.

Egy listát kiválaszthatunk, ha arra hosszan „kattintunk”.

### Lista létrehozása:

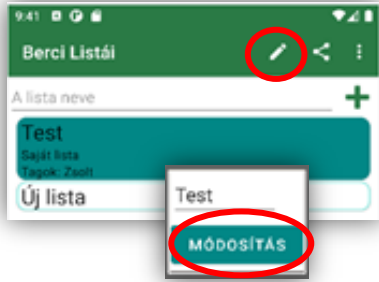
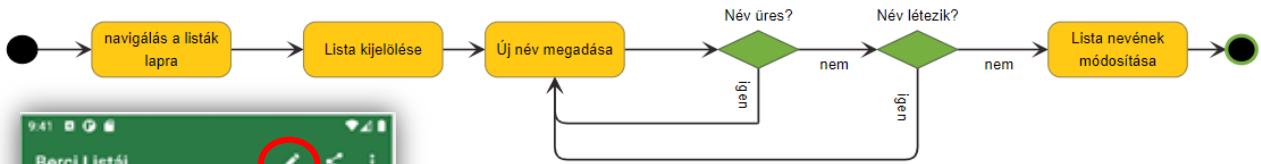


A felhasználó a listaválasztó képernyőn létrehozhat új listát. Az új lista nevének begépelése után a + gombra kattintva a felhasználó létrehozhatja az új listát. Az új lista nevét UI réteg és az adatkezelő réteg együttesen ellenőrzi. Megfelelő

(nem üres és még nem létező) név esetén új lista jön létre.

A gombra kattintva az UI ellenőrzi a megadott listanevet, hogy megfelel-e a formai követelményeknek. Ezt követően a névvel az adatkezelő réteg egy új dokumentumot hoz létre a felhasználó Lists kollekciójában.

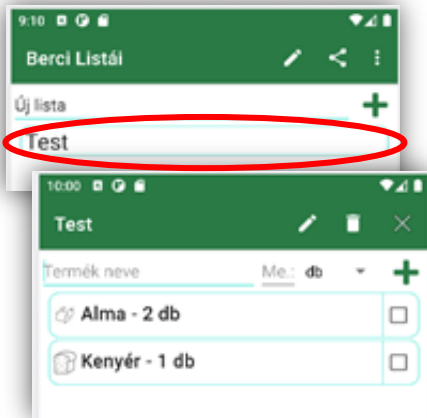
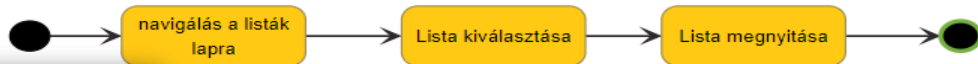
## Lista nevének szerkesztése:



A listanézetben a listát kijelölve szerkeszthetjük annak nevét. A név megadása után a módosítás gombra kattintva a név ellenőrzésre kerül, majd mentődik.

Az UI ellenőrzi, hogy a megadott név nem üres-e, majd a adatkezelő azonos nevű listákat keres. Ha az új név megfelel a kritériumoknak, akkor a módosítás végbemegy.

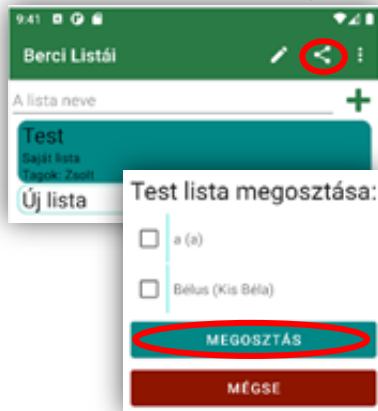
## Lista megnyitása:



A Listaválasztó nézetben lehetősége van a felhasználónak meglévő listát megjeleníteni. A Lista nevére kattintva megjelenik a választott lista listanézete.

A Lista nevére kattintva az UI lekéri az adatkezelő rétegtől választott lista termékeit és megjeleníti a kívánt listanézetet.

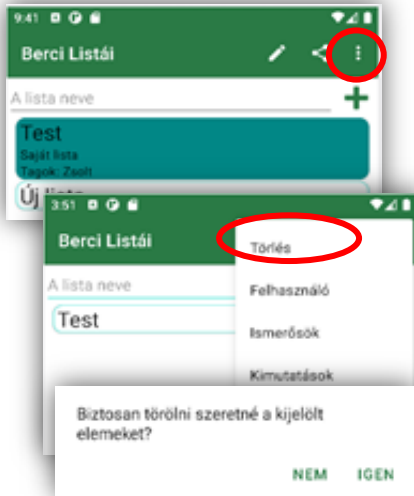
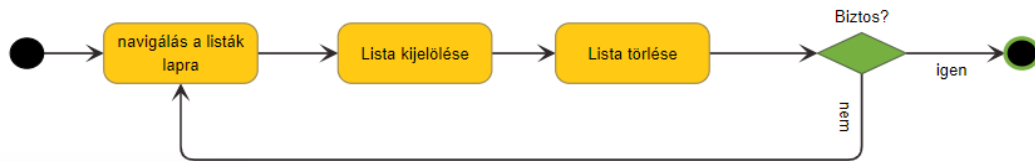
## Lista megosztása:



A listaválasztó nézetben, a lista kijelölése után, a megosztás gombra kattintva a listát megoszthatjuk a visszaigazolt ismerőseinkkel. A gombra kattintva az UI listázza az ismerőseinket. Ezek közül kiválaszthatjuk ki kapjon hozzáférést. A Lista dokumentumában a Shared, és SharedWith mezők módosulnak.

Az adatok megadása után adatkezelő réteg megkeresi az adott felhasználót a felhasználók kollekcióban. Az felhasználónál az external lists mezőben eltárolódnak a megosztott lista adatai.

## Lista törlése:



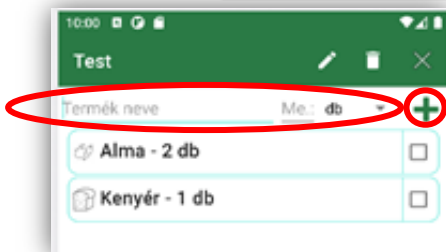
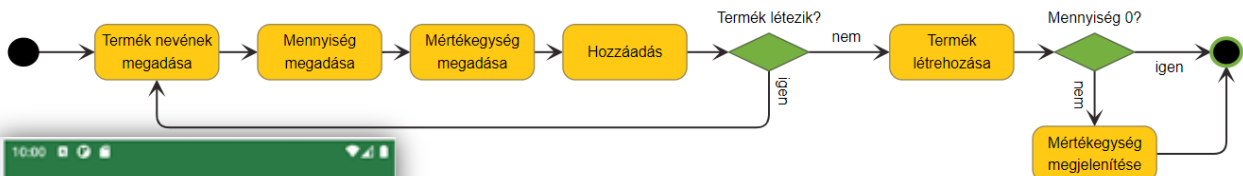
A listaválasztó képernyőn lehetőség van a lista törlésére. A törlésre kattintva az UI megerősítést kér. Megerősítés után a Lista eltűnik a felhasználó listái közül.

A törlés gombra kattintva az UI réteg továbbítja a lista kódját és a felhasználó kódját az adatkezelő rétegnek. Az ellenőrzi, hogy a felhasználó tulajdonosa-e a listának.

- Ha a felhasználó a lista tulajdonosa, akkor törli a listát a felhasználó listái közül, és minden tag ExternalList-jei közül.
- Ha a felhasználó nem tulajdonosa a listának akkor csak az adott felhasználó ExternalLists megfelelő értéke törlődik.

## 4.3. Termék manipuláló funkciók

### Termék listához adása:

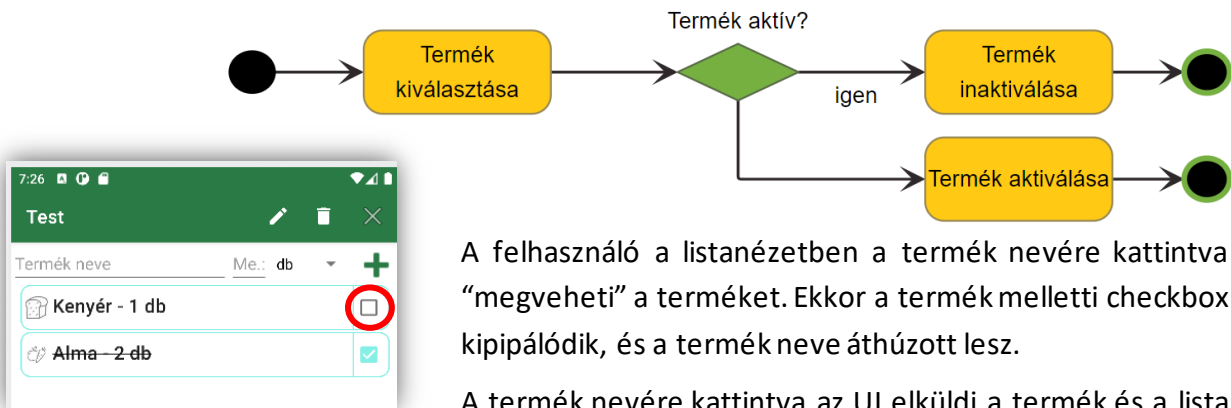


A listanézetben hozzáadhatunk egy terméket a megnyitott listához. A listára termék nevét begépelése közben egy lista megjeleníti a hasonló, korábban hozzáadott termékek neveit, ezekből választhatunk, vagy új nevet írhatunk be.

Ezután megadhatjuk a vásárolni kívánt mennyiséget, és annak mértékegységét (alapértelmezetten *db*). Ezek opcionális mezők, ha a felhasználó nem ad meg mennyiséget, akkor a mértékegység is üres marad.

Ezután a + gombra kattintva menthetjük az új terméket a listára. Ha ilyen nevű termék már létezik akkor az UI figyelmezteti a felhasználót és a folyamat megáll. Ha ilyen nevű termék még nincs a listán akkor a termék a megadott adatokkal hozzáadódik a lista products kollekciójának ProductsofList mezőjéhez, checked 0 értékkel.

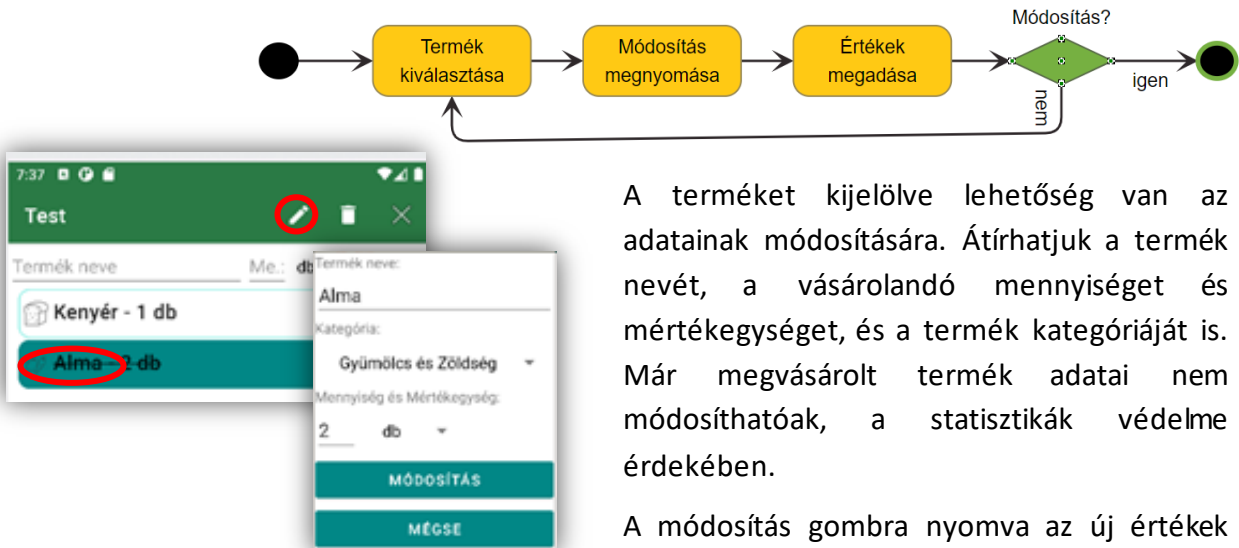
## Termék megvásárlása / vásárlás visszavonása:



A felhasználó a listanézetben a termék nevére kattintva “megveheti” a terméket. Ekkor a termék melletti checkbox kipipálódik, és a termék neve áthúzott lesz.

A termék nevére kattintva az UI elküldi a termék és a lista kódját az adatkezelő rétegnek, ami megkeresi azt a ProductsofList mezőben és a Cheched értékét 1-ra állítja. Hasonlóképpen, már megvett termék nevére kattintva visszavonhatjuk a vásárlást (újra felvehetjük a listára). Ekkor a Checked érték 0-ra változik.

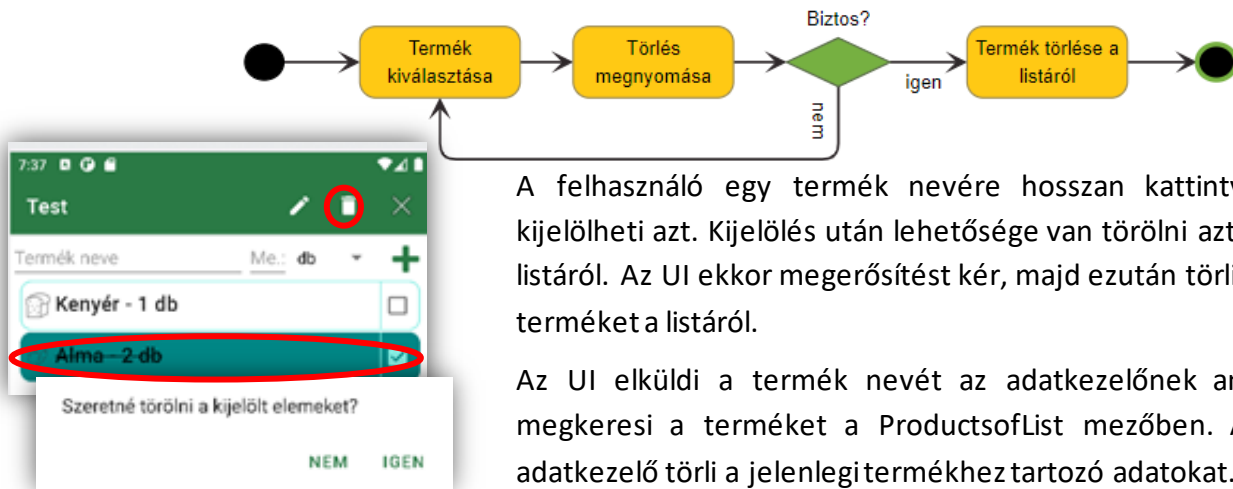
## Termék módosítása:



A terméket kijelölve lehetőség van az adatainak módosítására. Átírhatjuk a termék nevét, a vásárolandó mennyiséget és mértékegységet, és a termék kategóriáját is. Már megvásárolt termék adatai nem módosíthatóak, a statisztikák védelme érdekében.

A módosítás gombra nyomva az új értékek mentődnek.

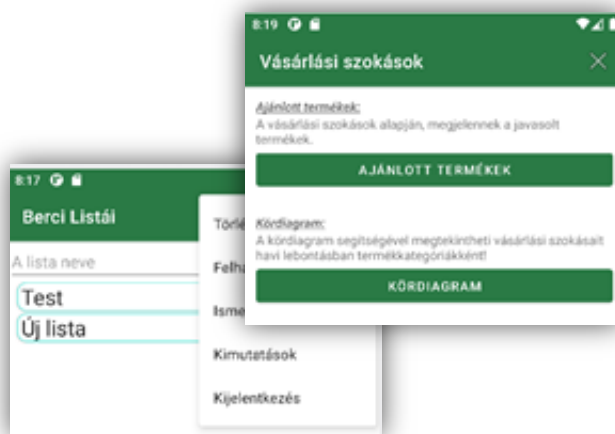
## Termék törlése a listáról:



A felhasználó egy termék nevére hosszan kattintva kijelölheti azt. Kijelölés után lehetősége van törölni azt a listáról. Az UI ekkor megerősítést kér, majd ezután törli a terméket a listáról.

Az UI elküldi a termék nevét az adatkezelőnek ami megkeresi a terméket a ProductsofList mezőben. Az adatkezelő törli a jelenlegi termékhez tartozó adatokat.

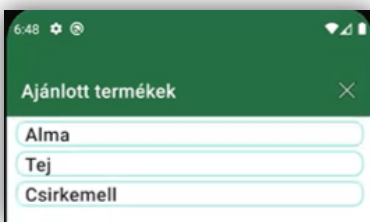
## 4.4. Kimutatások



A listaválasztó nézetből lehetőség van a kimutatások lapra navigálni, az egyek menüből. Ezen a lapon két további lehetőség jelenik meg: Az ajánlott termékek lap, illetve a Kördiagram lap.

Ezen a lapokon további információkat találunk a felhasználó vásárlási szokásaival kapcsolatban.

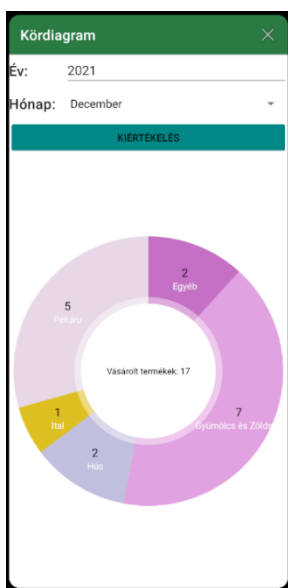
## Ajánlott termékek:



Az ajánlott termékek lapon láthatjuk azon termékeket, amelyeket viszonylag régen vásároltunk.

Az alkalmazás egy termék megvásárlásakor elmenti a vásárlás dátumát és kiszámolja az utolsó két vásárlás között eltelt időt. Ha ezt a terméket ennél régebben vásároltunk, akkor megjelenik az ajánlott termékek listáján.

## Statisztikák kérése:

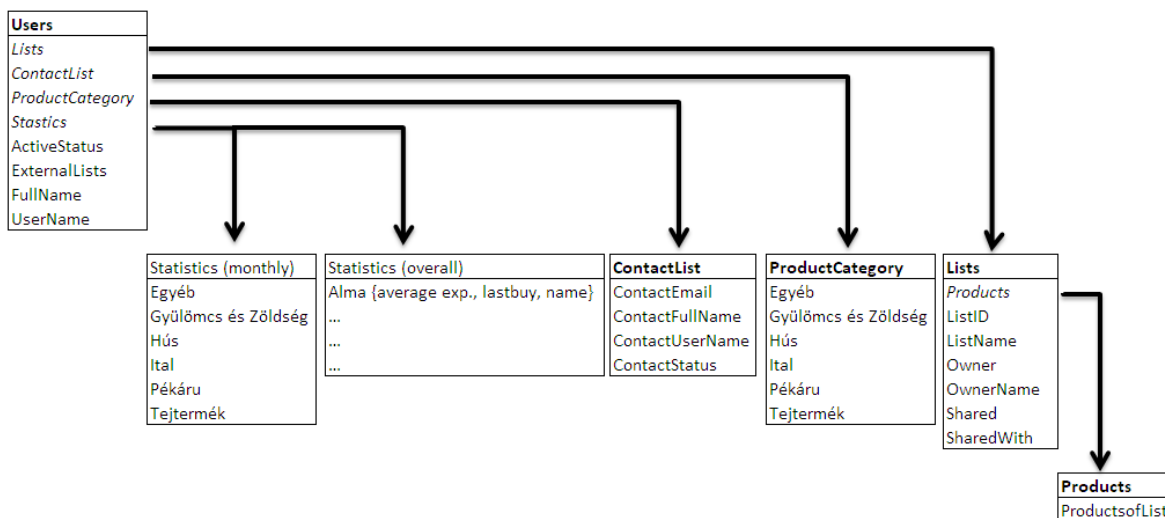


A Kördiagram képernyőn a felhasználó kiválaszthatja melyik hónap vásárlási összesítőjére kíváncsi.

A lapon megjelenik a termékkategóriák és a hónapban történt vásárlások számának kördiagramja.

## 5. Adatbázis felépítése

Egy Firestore adatbázisban úgynevezett kollekciókban dokumentumokat tárolunk. Egy dokumentumnak szintén lehet része kollekció, illetve további adatmezők. SQL adatbázisokkal párhuzamot vonva egy kollekció többé-kevésbé tekinthető egy táblának, míg egy dokumentum azon belül egy rekordnak.



Az adatbázisban az adatokat felhasználónként tároljuk. A *users* kollekció tartalmazza a regisztrált felhasználókat, azok listáit, ismerőseit, státuszát, illetve nevét. A felhasználó azonosítójaként E-mail címét használjuk.

Kollekció	Adatmezők	Típus			Leírás
Users	Lists	Kollekció			Saját listák kollekciója
	ContactList	Kollekció			„Ismerősök” kollekciója
	ProductCategory	Kollekció			Termékek és termékkategóriák
	Statistics	Kollekció			statisztikák kollekciója
	ActiveStatus	Boolean			Felhasználó státusza (aktív/törölt)
	ExternalLists	Array	ListID	String	felhasználóval megosztott listák Id-ja
			Owner	String	felhasználóval megosztott listák tulajdonosa
	FullName	String			Felhasználó teljes neve
	UserName	String			Megjelenítendő felhasználónév

A felhasználó ismerősei és listái szintén egy-egy kollekciót képeznek.

Kollekció	Adatmezők	Típus		Leírás
Users -> ContactList	ContactEmail	String		Kontakt Email címe
	ContactFullName	String		Kontakt teljes neve
	ContactStatus	Integer		Visszaigazolás státusza
	ContactUserName	String		Kontakt felhasználóneve

A felhasználók adatait több helyen is tároljuk. Ennek oka, hogy a használt Firestore adatbázis ingyenes verziója csak bizonyos számú olvasási műveletet enged naponta. A többszöri tárolással igyekszünk csökkenteni a query-k, és így az olvasások számát.

A termékeket a statisztikai kimutatások érdekében kategóriákba rendezzük, mint például „pékáru”, „zöldség és gyümölcsök”, stb. Ezeket a kategóriákat a felhasználó kollekcióján belül ProductCategory kollekcióban tároljuk. Egy termékkategória egy Array, amin belül a termékek nevét tároljuk.

Kollekció	Adatmezők	Típus		Leírás
Users -> ProductCategory	Egyéb	String Array		Egyéb termékek
	Gyümölcs és Zöldség	String Array		Gyümölcs és Zöldség termékek
	Hús	String Array		Hús termékek
	Ital	String Array		Ital termékek
	Pékáru	String Array		Pékáru termékek
	Tejtermék	String Array		Tejtermék



A felhasználó listáit külön, a felhasználón belüli kollekcióban tároljuk.

Kollekció	Adatmezők	Típus			Leírás
Users -> Lists	Products	Kollekció			Listán szereplő termékek
	ListID	String			Lista azonosítója
	ListName	String			Lista Neve
	Owner	String			Lista tulajdonosa
	OwnerName	String			Tulajdonos Neve
	Shared	Boolean			Nyilvánossági státusz
	SharedWith	Array,	ContactEmail	String	Listához hozzáférők adatai
			ContactFullName	String	
			ContactUserName	String	
			ContactStatus	String	

A Listákra felvett termékeket a listák kollekción belül egy újabb kollekcióban tároljuk. Ennek az oka, hogy így a listák beolvasásakor a termékeket még nem kérdezzük le az adatbázisból. Továbbá a termékeket egyetlen dokumentum Array típusú mezőjében tároljuk, így csökkentve a dokumentumok olvasásának számát.

Kollekció	Adatmezők	Típus			Leírás
Users -> Lists -> Products	ProductsofList	Array	Checked	Boolean	Kell-e még venni a termékből
			Name	String	Termék neve
			Quantity	String	Vásárolandó mennyiség
			QuantityType	String	Mértékegység

A statisztikák kollekcióban a felhasználó vásárlási szokásait tároljuk kétféle dokumentumban. Az egyik dokumentumtípus az összesített statisztikákat tárolja, és a termékek ajánlásához nyújt információt. Ez a dokumentum termékenként tárolja az utolsó vásárlás dátumát, a termék nevét, és az utolsó két vásárlás között eltelt napok számát.

Egy termék megvásárlásakor a az új és a régi lastBuyDate különbsége eltárolódik az average mezőben.

Ha az utolsó vásárlás óta eltelt ugyanennyi nap, akkor a termék felkerül az ajánlott termékek listára.

Kollekció	Adatmezők	Típus			Leírás
Users -> Statistics/ buyStatistics	Alma	Array	average	String	utolsó két vásárlás között eltelt napok száma
	...		lastBuyDate	String	utolsó vásárlás dátuma
	...		name	String	termék neve

A másik dokumentumtípus havonta összegzi az egyes termékkategóriákhoz rendelhető vásárlások számát. Ez az adat a kördiagram készítéséhez használatos. Egy termék megvásárlásakor az adott termék kategóriáján belüli mennyiség 1-el nő.

Kollekció	Adatmezők	Típus			Leírás
Users -> ProductCategory	Egyéb	map	Quantity	number	Az adott termékkategóriában, adott hónapban történt vásárlások száma.
	Gyümölcs és Zöldség	map	Quantity	number	
	Hús	map	Quantity	number	
	Ital	map	Quantity	number	
	Pékáru	map	Quantity	number	
	Tejtermék	map	Quantity	number	

## 6. Tesztek

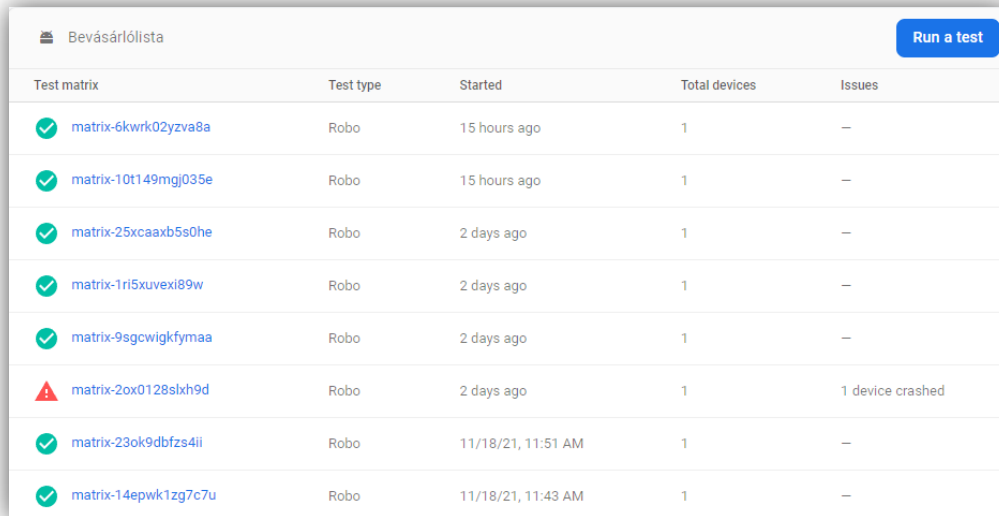
Az alkalmazást a fejlesztés több szakaszában is alávetettük manuális, illetve a Firestore kínálta Test Lab automatikus tesztjeinek is.

A manuális tesztek során vizsgáltuk az egyes funkciók működését, vagyis, hogy az egyes funkció használata során kapott eredmények megfelelnek-e az elvártnak. A tesztek tapasztalatait felhasználva próbált változtatásokat vezettünk be. Például:

- Már megvásárolt termék neve kategóriája nem szerkeszthető, mert ez a kimutatásokban váratlan értékeket okoz.
- Lista Névszerkesztő felugró ablak kiegészült egy “mégse” paranccsal
- Ismerősnek jelölés elfogadása/elutasítása ablak kiegészült egy “mégse” paranccsal

Funkcionális tesztek mellett végeztünk teljesítményteszteket is. Ezekből például kiderült, hogy egyes esetekben a Firestore online adatbázis elérése lassú lehet. Annak érdekében, hogy gyenge internetkapcsolat esetén az applikáció inkább az offline adatbázist használja, beépítésre került egy kapcsolat figyelő modul, amely a kapcsolat erősségétől függően engedélyezi/tiltja az online adatbázis elérését.

A manuális teszteken kívül a Firestore Test Lab automatikus tesztelési lehetőségeit is kihasználtuk. Az automatikus tesztek videofelvételen dokumentálva lettek.



Test matrix	Test type	Started	Total devices	Issues
matrix-6kwrk02yzva8a	Robo	15 hours ago	1	—
matrix-10t149mgj035e	Robo	15 hours ago	1	—
matrix-25xcaaxb5s0he	Robo	2 days ago	1	—
matrix-1ri5xuvexi89w	Robo	2 days ago	1	—
matrix-9sgcwigkfymaa	Robo	2 days ago	1	—
matrix-2ox0128slxh9d	Robo	2 days ago	1	1 device crashed
matrix-23ok9dbfzs4ii	Robo	11/18/21, 11:51 AM	1	—
matrix-14epwk1zg7c7u	Robo	11/18/21, 11:43 AM	1	—

Ahogy a képen látható egy teszt során az applikáció összeomlott. Ennek oka a kapcsolati állapotot figyelő ConnectivityManager (CM) az aktivitási életciklus nem megfelelő életszakaszában való lekapcsolása, amiről a teszt eredmények közlése során egyértelmű üzenetet kaptunk:

#### Fatal exception

```
java.lang.RuntimeException: Unable to pause activity {com.example.recyclerview2/com.example.recyclerview2.lists.ListActivity}:
java.lang.IllegalArgumentException: NetworkCallback was already unregistered
```

A CM az onPause() metódus/életciklus alatt került leválasztásra, amint új aktivitást nyitott meg a teszt (jelen esetben egy listát), visszatérést követően ez nem okozott azonnal problémát, csak amikor újra megnyitott egy listát, akkor a már létező kapcsolatot onPause() ciklus elérésekor értelemszerűen nem tudta megszüntetni. A megoldást az jelentette, hogy a kapcsolat bontása átkerült az onDestroy ciklusba, ahol a helye eredetileg is helyes lett volna, mivel ezt követően az aktivitás csak az onCreate ciklussal térhet vissza a képernyőre, és ebben az életszakaszban kerül a CM networkcallback regisztrálásra.

A felvételek alapján az alkalmazás az elvárt módon reagált minden esetben. Azonban megjegyzendő, hogy az automatizmus nem minden esetben ismerte fel a beviteli mező típusát. Regisztráció tesztelésekor ez problémákat okozott. Ez feltárta bizonyos beviteli mezők megfelelő azonosíthatóságának hiányát. Ezt javítva, és a megfelelő beviteli értékeket megadva az automatizmus is képes volt a regisztrációs folyamat tesztelésére. A javítás utáni teszt megfelelő eredménnyel zárult.

Az automatikus tesztek szintén rávilágítottak, hogy bizonyos helyeken (3) "Non-SDK interfészeket" használtunk, amelyek az android környezettől függetlenül változhatnak, és így később kompatibilitási problémákat okozhatnak egyes android verziók esetén.

<code>Ljava/security/spec/ECParameterSpec;-&gt;setCurveName(Ljava/lang/String;)V</code>
<code>Ljava/net/Socket;-&gt;impl:Ljava/net/SocketImpl;</code>
<code>Ljava/security/spec/ECParameterSpec;-&gt;getCurveName()Ljava/lang/String;</code>

A tesztek alapján elmondható, hogy az alkalmazás megfelel a csapat elvárásainak. A jövőben a fenti "Non-SDK interfész" figyelmeztetés javításán kívül, apróbb kényelmi funkciók beépítését látjuk relevánsnak.