



2025

# Grocery List



Csákó Balázs

2025.01.28.

IOMAG5

# Bevezetés

**Grocery List** egy bevásárlólista készítő alkalmazás, amivel egyszerűen és gyorsan lehet felvenni, törölni és módosítani bejegyzéseket. Valamint az appból való kilépés során sem veszik el az addig létrehozott bejegyzések.



Az alkalmazás Kotlin programozási nyelven készült, és Android Studióban lett megvalósítva. Sajnálatos módon kompatibilitási gondok miatt kénytelen voltam a Studió egy újabb verzióját használni. Konkrétan az Android Studió Ladybug 2024.02.01 környezetben a Gradle 8.8-al egyetemben.

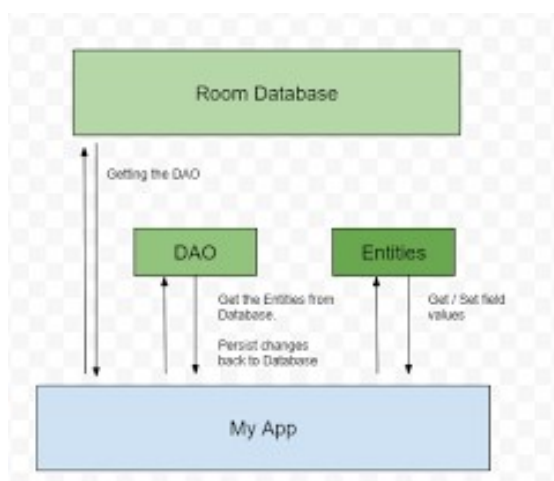


Maga az alkalmazás az Android 1.14 (Sdk 34), avagy Upside Down Cake-re készült, emellett a Build folyamat során elengedhetetlen volt a Java is. A fejlesztés során fontos szerepet játszott a kotlin-kapt (Kotlin Annotation Processing Tool) modul is, amit helyett használhattam volna az újabb KSP-t (Kotlin Symbol Processing). Azonban néhány függőség ezt megakadályozta.

## Adattárolás

Androidban több megoldás is létezik az adatok tárolására. Ilyen például SharedPreferences, File Storage, Firebase vagy DataStore is. Azonban az alkalmazásban Room Database 2.6.1-os verzióját használtam, ami egy SQLite alapú rendszer. Ennek előnye, hogy az Android Jetpack része, ebből kifolyólag könnyen integrálható.

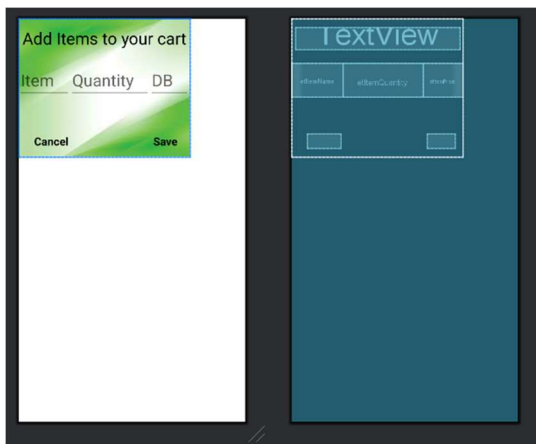
Ennel a rendszernek elengedhetetlen része a Dao, amely az adatbázis és az alkalmazás közötti kommunikációt biztosítja. A DAO szerepe, hogy meghatározza az adatbázisban végzett műveleteket (pl. lekérdezések, beszúrás, frissítés, törlés). Másik fontos elem az Entity, amely a tárolt objektumot határozza meg. Jelen esetben, az alkalmazásban



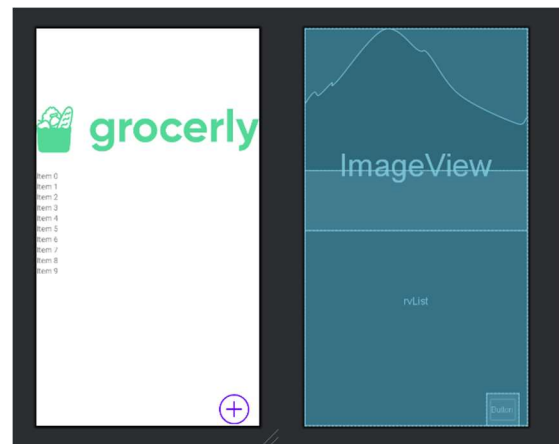
3 módosítható és egy rejtett tulajdonsága van az Entity-nek. Ez a vásárolni kívánt tárgy neve, annak mennyisége (egész szám) és annak kiszámlázása (db, csomag, stb.), valamint az objektum egyedi ID-je, amit a rendszer automatikusan generál. Ezen ID segítségével történik a módosítás és törlés is.

## Design

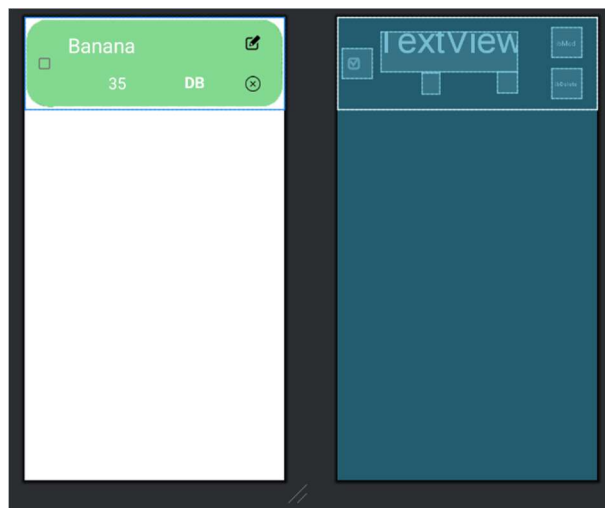
Az alkalmazás során mindössze 3 különböző layoutot használtam, amik az *activity\_main.xml*, a *dialog.xml* és az *itemadapter.xml*. Míg a használt színeket és szövegeket a *colors* és *strings.xml* tartalmazza. Leginkább a zöld árnyalatait, a fehér és fekete színeket használtam.



dialog.xml



activity\_main.xml

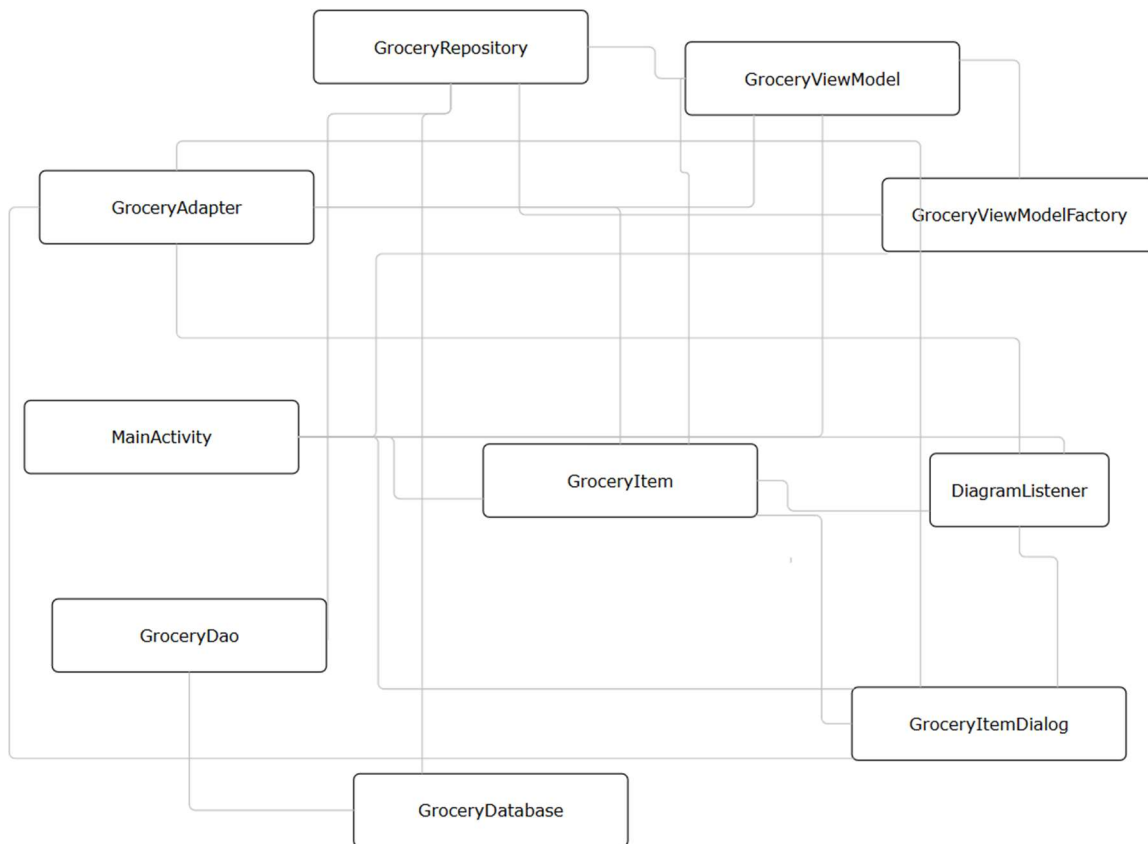


itemadapter.xml

Az *activity\_main* igazából a szoftver kerete, amely tartalmazza a többi objektumot. Fontos része a RecyclerView, amely nagy mennyiségű, görgethető adathalmaz hatékony megjelenítésére szolgál. Ez igazából a ListView modernebb, rugalmasabb és optimalizált változata, amely jobb teljesítményt és testreszabhatóságot kínál. Valamint az új objektumok hozzáadására szolgáló gomb is itt található meg.

A *dialog* az új elemek adatainak bevitelére szolgál, annak 3 EditText Field-jeivel. Míg az *itemadapter* ezen adatbázisba felvitt objektumok megjelenítésére. Emellett ez tartalmaz egy checkboxot is, ami elengedhetetlen része egy bevásárló listának. Valamint ez az xml tartalmazza az adott elem törlését és módosítását lehetővé tevő gombokat.

## Projekt struktúra

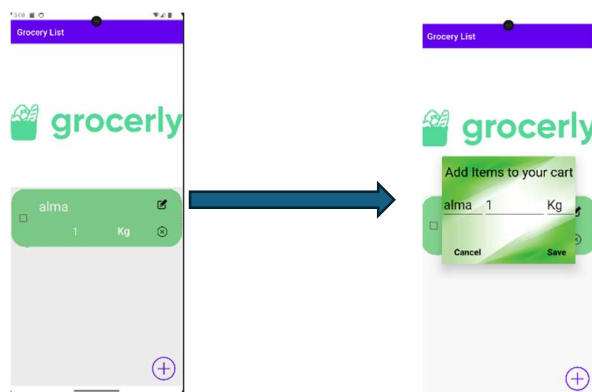


# Alkalmazás használata

Új objektum, felvételéhez a job alsó sarokban lévő gombra kattintva lehet. A gombra történő kattintás után megjelenik egy ablak, amibe az tárgy Adattárolás fejezetben említett tulajdonságait lehet megadni. Viszont a név az kötelező, annak hiányában az alkalmazás szól. A mennyiség alapértelmezett értéke 1, a harmadiknak pedig darab. A Save-re kattintva menti, míg a Cancel eldobja.



Módosításkor a tárgy jobb oldalán található felső gombra kattintva megint megjelenik a felvételkor látott ablak, azonban ez a tárgy adataival már ki van töltve. Itt is ugyanúgy a Save-re kattintva menti.



Törölni a tárgy jobb lent található X gombbal lehet

# **Források**

Órai anyag

[GeeksforGeeks](#)

[Android Developers](#)

[Stackoverflow](#)

[Kotlin Documentation](#)

[GitHub Forum](#)

[JohnCodes](#)