# Mobil programozás alapok
# beadandó feladat jegyzőkönyv

WishList App  mobilalkalmazás

2025. 01. 06

Pázmán András

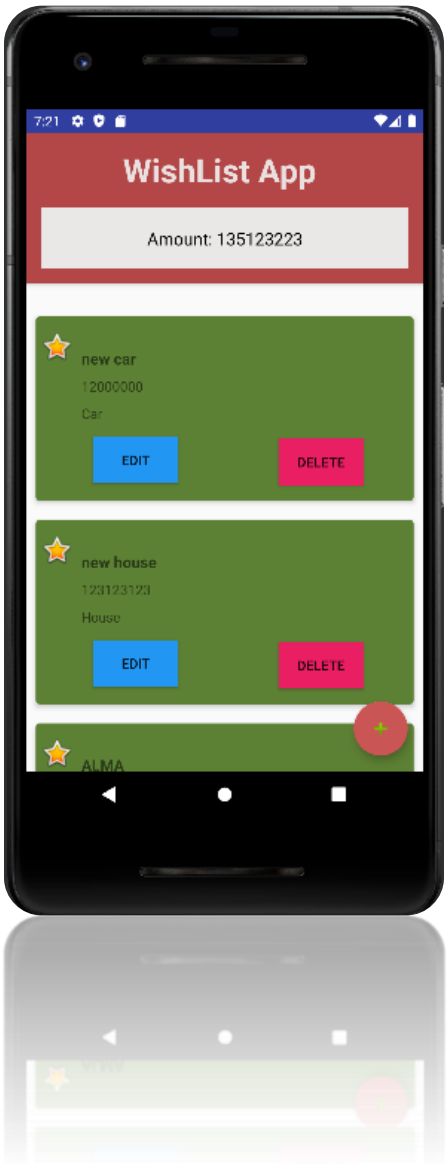
H2Z4X3

**Feladat ismertetése**

Nyilvántartó  alkalmazás készítése amely segítségével egy kívánságlístát állíthatunk össze magunknak. A lista segítségével kategorizálni tudjuk az adott itemet melyhez összeget is rendelhetünk.
Az adott tételeket az app összegzi és megjelenití.
A felvételen kívül lehetőségünk van módosítani és törölni is a felvett elemeket.

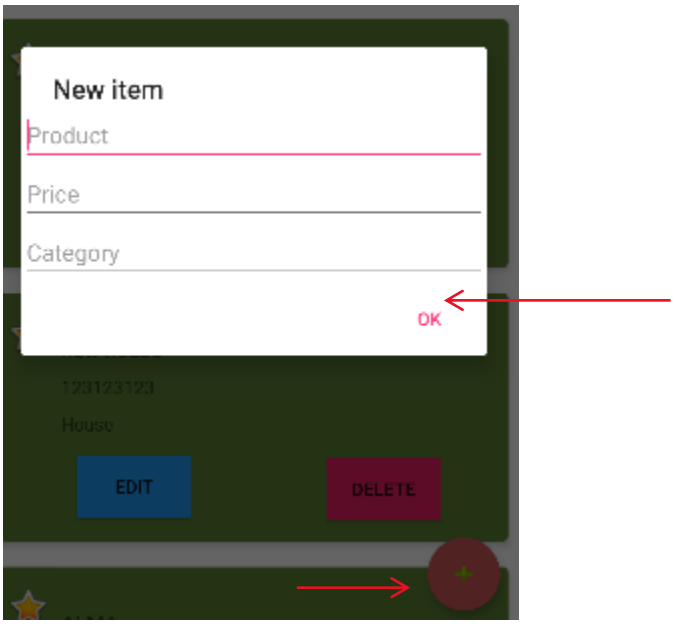
A feladat megvalósításán android studio-ban dolgoztam Kotlin nyelvvel .

**Az alkalmazás felülete:**



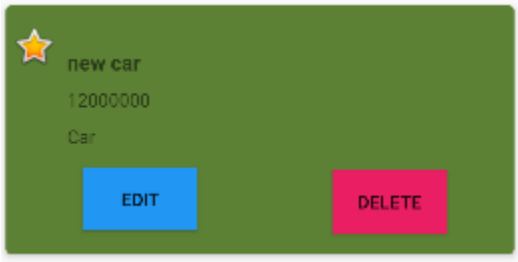
**Kezelőfelület funkciói:**

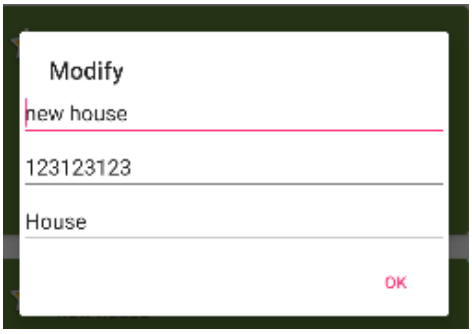
Új elem hozzáadása:




A “ + “ gomb megnyovásával  ugrik fel a hozzáadás panel. A megfelelő mezőket kitöltve tudunk új elemet felven a listába.

A kártyán megjelenő felvett elemet könnyen tudjuk ezután módosítani az “Edit “ gombbal.



**Adatbázis létrehozása Room-mal**
Amikor az alkalmazásban egy adatbázist hozol létre a Room segítségével, akkor az **Amounts** osztály alapján a Room automatikusan létrehozza az **Amounts** nevű táblát.

**Hogyan működik az adatbázis műveletek során?**

1. **Beszúráskor:** A Room az **amounts** táblába egy új sort szúr be az Amounts osztály alapján. Az amountId automatikusan generálódik.
2. **Lekérdezéskor:** A Room az **Amounts** osztály példányait tölti be az adatbázisból.
3. **Frissítéskor:** A Room a megadott oszlopokat módosítja.
4. **Törléskor:** A Room azonosítók alapján eltávolítja a sort a táblából.

```kotlin
@Entity(tableName = "amounts")
data class Amounts(
    // Primary key for the table, automatically generated.
    @PrimaryKey(autoGenerate = true) var itemID: Long?,

    // Column to store the description of the expense.
    @ColumnInfo(name = "description") var description: String,

    // Column to store the amount of the expense.
    @ColumnInfo(name = "amount") var amount: Int,

    // Column to store the category of the expense.
    @ColumnInfo(name = "category") var category: String
) : Serializable // Makes the class serializable for easier data transfer.
```

```kotlin
interface DatabaseOps {

    // Retrieves a list of all expenses from the "expenses" table.
    @Query( value: "SELECT * FROM expenses")
    fun findAllExpenses(): List<Amounts>

    // Inserts a new expense into the "expenses" table.
    // Returns the autogenerated primary key of the inserted row.
    @Insert
    fun insertExpense(amounts: Amounts): Long

    // Deletes an expense from the "expenses" table.
    // The specific row is identified based on the values of the provided Amounts object.
    @Delete
    fun deleteExpense(amounts: Amounts)

    // Updates an existing expense in the "expenses" table.
    // The row to update is identified by the primary key of the provided Amounts object.
    @Update
    fun updateExpense(amounts: Amounts)

    // Calculates the total sum of the "amount" column in the "expenses" table.
    // Useful for summarizing all expenses.
    @Query( value: "SELECT SUM(amount) FROM expenses")
    fun getTotalExpense(): Int
}
```