

Mobil Programozási alapok
KotlinShoppingList feladat átdolgozása

Workout app létrehozása

Név: Kiss Máté

Neptunkód: NXYCH1

Workout App létrehozása

A KotlinShoppingList android studios projekt alkalmazáshoz hasonlóan alkottam meg a Workout App alkalmazást.

A KotlinShoppingList project mappában dolgoztam, a refactor nem engedi a root név átírását!

Az alkalmazás működik, nem tartalmaz hibát.

Funkciói: Új Workout felvétele, Meglévő Workout szerkesztése, Meglévő Workout törlése.

A rögzített Workoutok egy SQL adatbázisban tárolva vannak. A feladathoz Az Android Studio alap Medium Phone API 35 ös virtuális telefont használtam.

Az Andorid Studio 2021.1.1-es verziójával írtam a programot.

1. Adatszerkezet: WorkoutItem az edzéssel kapcsolatos adatok tárolására
Tárolt adatok:

itemID, automatikusan generálva,

name az edzés gyakorlat neve,

repeat: ismétlési darabszám,

check: egy bepípálható checkbox

sets: hányszor kell újra ismételni az adott gyakorlatot és darabszámot,

resttime: pihenő idő a sets ek között.

```
import ...

/*
    Adatbázis táblát készíti el.
    Táblanév:workoutitem.
    Oszlopok:itemId, name, repeat, check, sets, resttime.
    @PrimaryKey(autoGenerate = true): elsődleges kulcs, automatikusan generálva.
    Ide szükséges a bővítés új adattal.
 */
@Entity(tableName = "workoutitem")
data class WorkoutItem(@PrimaryKey(autoGenerate = true) var itemId: Long?,
    @ColumnInfo(name = "name") var name: String,
    @ColumnInfo(name = "repeat") var repeat: Int,
    @ColumnInfo(name = "check") var check: Boolean,
    @ColumnInfo(name = "sets") var sets: String,
    @ColumnInfo(name = "resttime") var restTime: String
) : Serializable
```

2. Adatbázis: AppDatabase

Az AppDatabase egy Room adatbázis implementációt csinál az androidos alkalmazáshoz a workout.db hez. a WorkoutItem entitás alapján tárol adatokat. az AppDatabase kiterjeszti a RoomDatabase osztályt.

3. WorkoutItemDao: A DAO osztály, aminek a feladata az adatbázis kezelése.

Az általa végzett metódusok:

findAllItems() Az összes elem kilistázása

insertItem Egy elem behelyezése

deleteItem Egy elem törlése

updateItem Egy elem frissítése

4. A program működéséért és megjelenítéséért a MainActivity felel.

A kód feladata a felhasználói felület kezelésére a MainActivity, amely lehetővé teszi az edzésadatok (Workout Item-ek) megjelenítését és a különböző műveletek végrehajtását, például új edzésadatok hozzáadását, módosítását és törlését. Az alkalmazás felhasználói felületének központi eleme az edzéslista, amely egy RecyclerView komponens segítségével jelenik meg, és az elemeket egyedi kártyákra rendezi. A felhasználók az új elemek hozzáadását egy gomb segítségével végezhetik el, miközben a meglévő elemek módosítása és törlése érintésalapú interakcióval is lehetséges.

5. Az XML fileok

A program megjelenéséért a layout mappa xml filejai felelősek.

Az **activity_main.xml** file a MainActivity megjelenítését tartalmazza.

Az egyedi gomb a jobb alsó sarokban egy ImageView al lett hozzáadva mivel a android studio nem rendelkezett olyan ikonnal amit megfelelőnek találtam volna a feladatra.

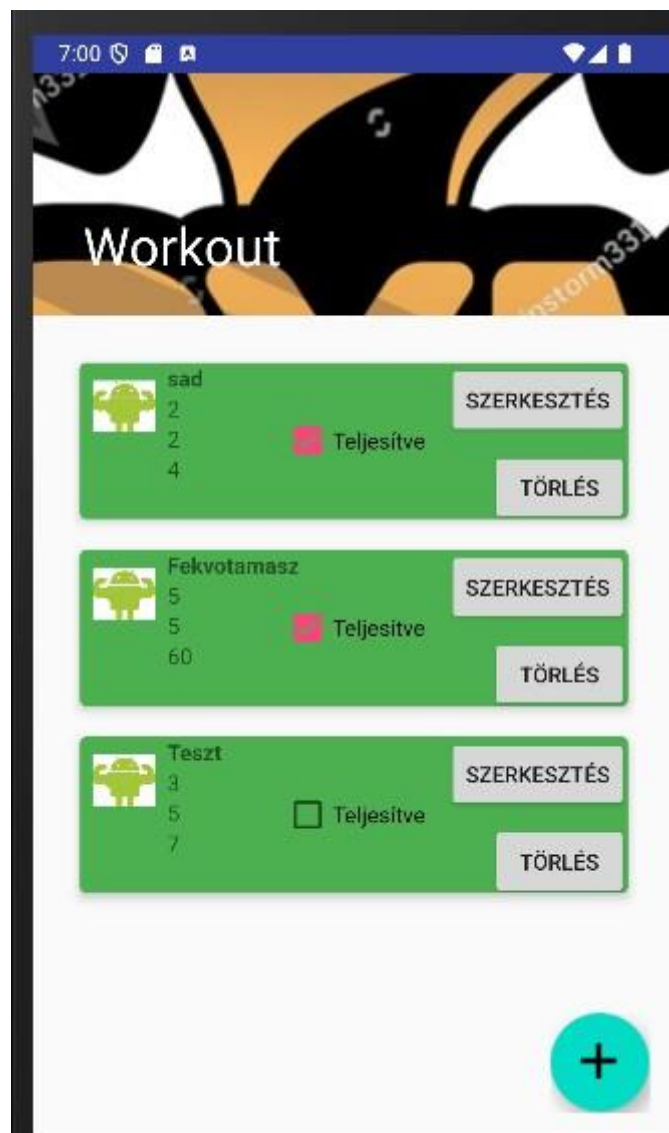
```
<ImageView
    android:id="@+id/add_button"
    android:layout_width="64dp"
    android:layout_height="64dp"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:src="@drawable/plus"
    android:contentDescription="Add Item"
    android:background="@android:color/transparent"
    android:clickable="true"
    android:focusable="true"/>
```

A **dialog_create_item.xml** a felugró ablakért felel, amiben az új elemeket felvisszük.

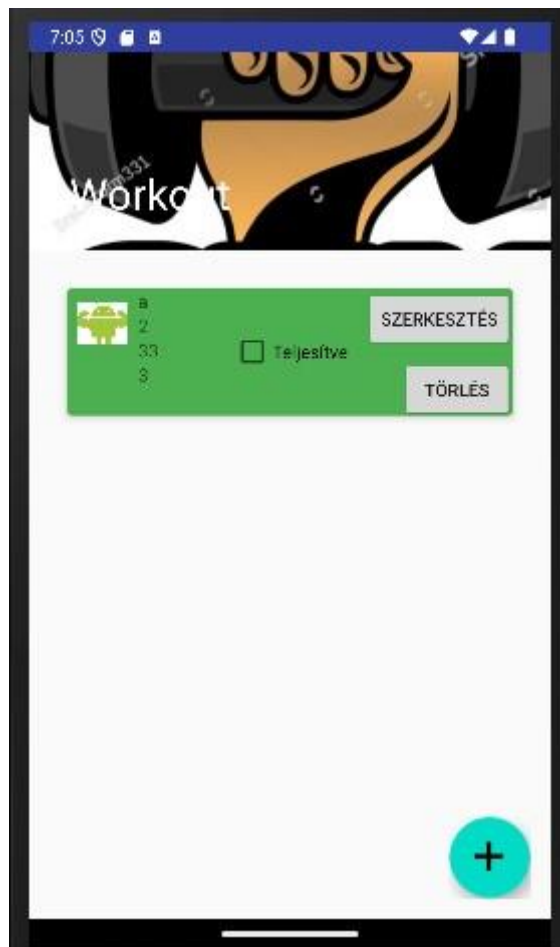
A **row_item.xml** felel a kártya elemen történő megjelenítésért.

A design a drawable mappában található képeket használja fel.

6. Képek a működésről:



Létrehozott kártyák



A Szerkesztés és törlés működése

Új Gyakorlat

OK

kártya felvitel