Beadandó feladat
Shopping List

Bodnár László
D1H8VP

A beadandóm célja egy kotlin nyelven irt Android alkalmazás fejlesztése, ami egy bevásárlólistát kezel.

A feladatot Android Studioban oldottam meg, és JetPack compose Ui-t használtam, az architektúrám a standard Model-View-Viewmodel. Adatbázisként az SQL-liteot használtam Room ORM-segítségével, az adatbázist a telefonon tárolom.

Az adatbázisomért a ShoppingItem.kt file felel és a következő mezőket tartalmazza:
id: Elsődleges kulcs
name: Termék neve
quantity: Mennyiség

isChecked: Logikai érték, a termék létét vizsgálja a kosárba

```
@Entity(tableName = "shopping_table")
data class ShoppingItem(
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,
    val name: String,
    val quantity: Int,
    val isChecked: Boolean = false
)
```

Az adatelérésért a ShoppingDao.kt felel.

A Dao egy Data Access Object interfész amely SQL-lekérdzéseket tartalmaz.

```kotlin
@Dao
interface ShoppingDao {

    1 Implementation
    @Query( value = "SELECT * FROM shopping_table WHERE name LIKE '%' || :searchQuery || '%' ORDER BY isChecked ASC, id DESC")
    fun getItems(searchQuery: String): Flow<List<ShoppingItem>>

    1 Usage  1 Implementation
    @Insert(onConflict = OnConflictStrategy.IGNORE)
    suspend fun insert(item: ShoppingItem)

    1 Usage  1 Implementation
    @Update
    suspend fun update(item: ShoppingItem)

    1 Usage  1 Implementation
    @Delete
    suspend fun delete(item: ShoppingItem)


    1 Implementation
    @Query( value = "DELETE FROM shopping_table")
    suspend fun deleteAll()


    1 Implementation
    @Query( value = "DELETE FROM shopping_table WHERE isChecked = 1")
    suspend fun deleteCompleted()
}
```

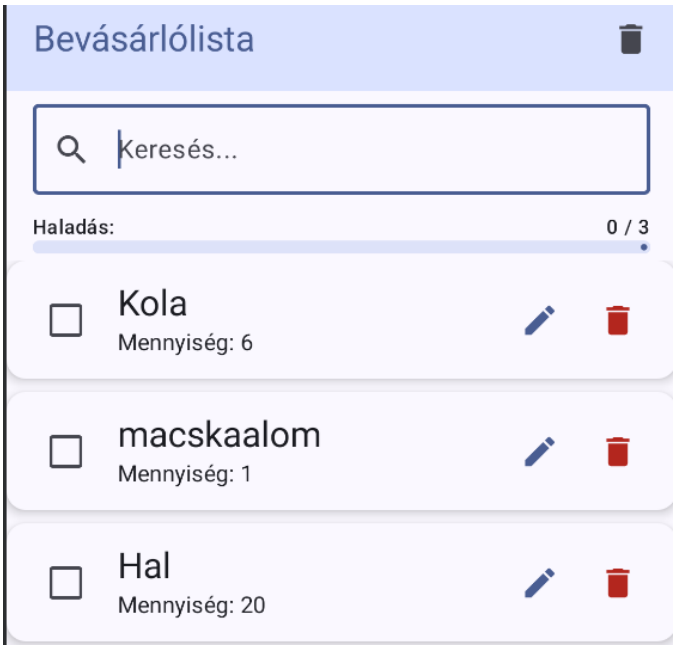
A viewmodel A ShoppingViewModel osztályon belül kerül megvalósításra.

A keresés a flatMapLatest operátor segítségével történik.

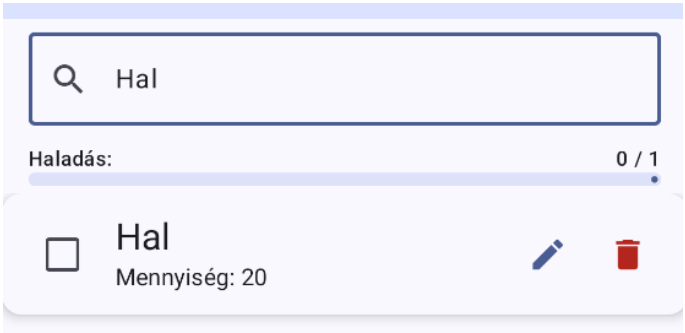
```kotlin
val items = _searchText.flatMapLatest { query ->
    dao.getItems( searchQuery = query)
}.stateIn(viewModelScope, started = SharingStarted.WhileSubscribed( stopTimeoutMillis = 5000), initialValue = emptyList())
```

Az applikációban ez igy látszódik:

Keresetlen állapot:



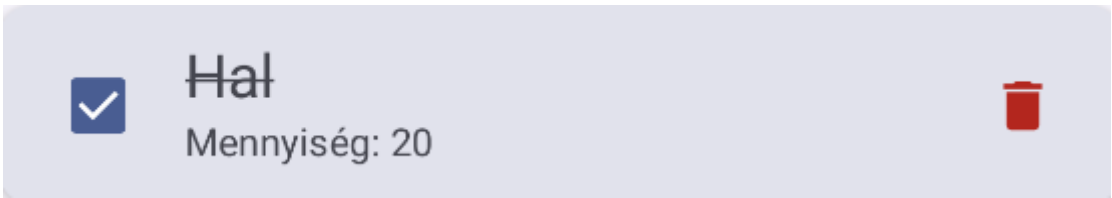
Keresett állapot:



A lista megjelenítéséért a LazyColumn komponens felel, ami hatékony memóriakezelést tesz lehetővé.

Extraként bekerült egy gomb, amit ha kipipálunk az adott terméket teljesítettnek veszi, ha készként van jelölve egy termék nem lehet módosítani se nevét se tartalmat.

```
if (!item.isChecked) {
    IconButton(onClick = onEdit) {
        Icon( imageVector = Icons.Default.Edit, contentDescription = "Szerkesztés", tint = MaterialTheme.colorScheme.primary)
    }
}
```

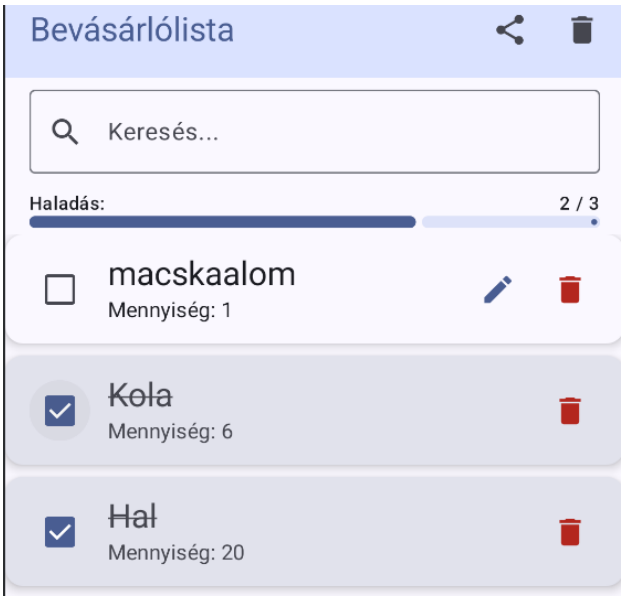


A bevásárló lista ezeken kivul természetesen tartalmazza az alapfunkciókat pl a törlés és szerkesztés.

A törlés során lehetőségünk vagy egy elemet, vagy az összeset törölni.

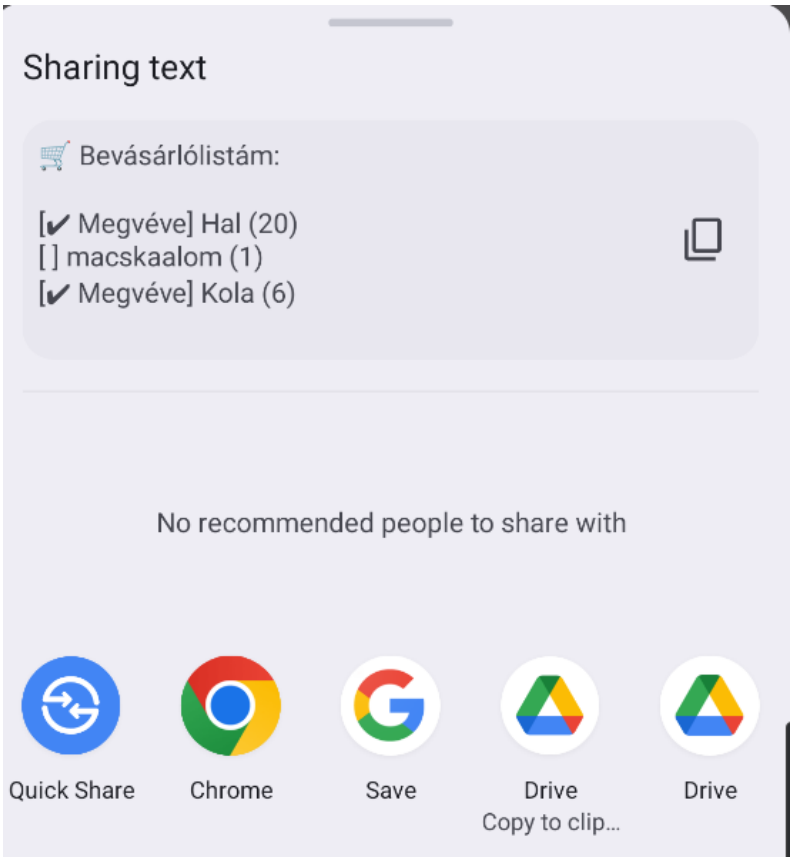
A fenti fekete mindent töröl, a termék melletti piros pedig csak az alábbi terméket.

A bevásárló listám tartalmaz egy Haladás mérőt, amely megmutatja, hogy x termékből mennyi van már a kosárba.




```
if (totalItems > 0) {
    Column(modifier = Modifier.padding(horizontal = 16.dp, vertical = 4.dp)) {
        Row(
            modifier = Modifier.fillMaxWidth(),
            horizontalArrangement = Arrangement.SpaceBetween
        ) {
            Text( text = "Haladás:", style = MaterialTheme.typography.labelMedium)
            Text( text = "$completedItems / $totalItems", style = MaterialTheme.typography.labelMedium)
        }
        LinearProgressIndicator(
            progress = { progress },
            modifier = Modifier.fillMaxWidth().height( height = 8.dp),
```

Végül az applikáció tartalmaz egy megosztás funkciót, hogy más készülékekkel, családtagokkal meg tudjuk osztani az összeállított Listát.




```
Scaffold(
    topBar = {
        Column {
            TopAppBar(
                title = { Text( text = "Bevásárlólista") },
                actions = {

                    IconButton(onClick = { shareList(context, allItems) }) {
                        Icon( imageVector = Icons.Default.Share, contentDescription = "Megosztás")
                    }

                    IconButton(onClick = { viewModel.deleteAllItems() }) {
                        Icon( imageVector = Icons.Default.Delete, contentDescription = "Összes törlése")
                    }
                },
                colors = TopAppBarDefaults.topAppBarColors(
                    containerColor = MaterialTheme.colorScheme.primaryContainer,
                    titleContentColor = MaterialTheme.colorScheme.primary
                )
            )
```

```
fun shareList(context: android.content.Context, items: List<ShoppingItem>) {

    val text = StringBuilder()
    text.append("🛒 Bevásárlólistám:\n\n")

    items.forEach { item ->
        val status = if (item.isChecked) "[✓ Megvéve]" else "[ ]"
        text.append("$status ${item.name} (${item.quantity})\n")
    }

    val sendIntent = android.content.Intent().apply {
        action = android.content.Intent.ACTION_SEND
        putExtra( name = android.content.Intent.EXTRA_TEXT,  value = text.toString())
        type = "text/plain"
    }

    val shareIntent = android.content.Intent.createChooser( target = sendIntent,  title = "Lista küldése...")
    context.startActivity(shareIntent)
}
```