

# Grocery App Report

## **1. Introduction**

### **1.1 Overview**

This is an android app that helps you to make a list of grocery items along with its price and quantity.

### **1.2 Purpose**

We are humans and we cannot remember everything. We sometimes forget the things that we want to buy. However, with the assistance of this app you can make a list of grocery items you intend to buy so that you don't forget anything and also have a track of your expenditure for budget maintenance.

## **2. Literature Survey**

### **2.1 Existing Problem**

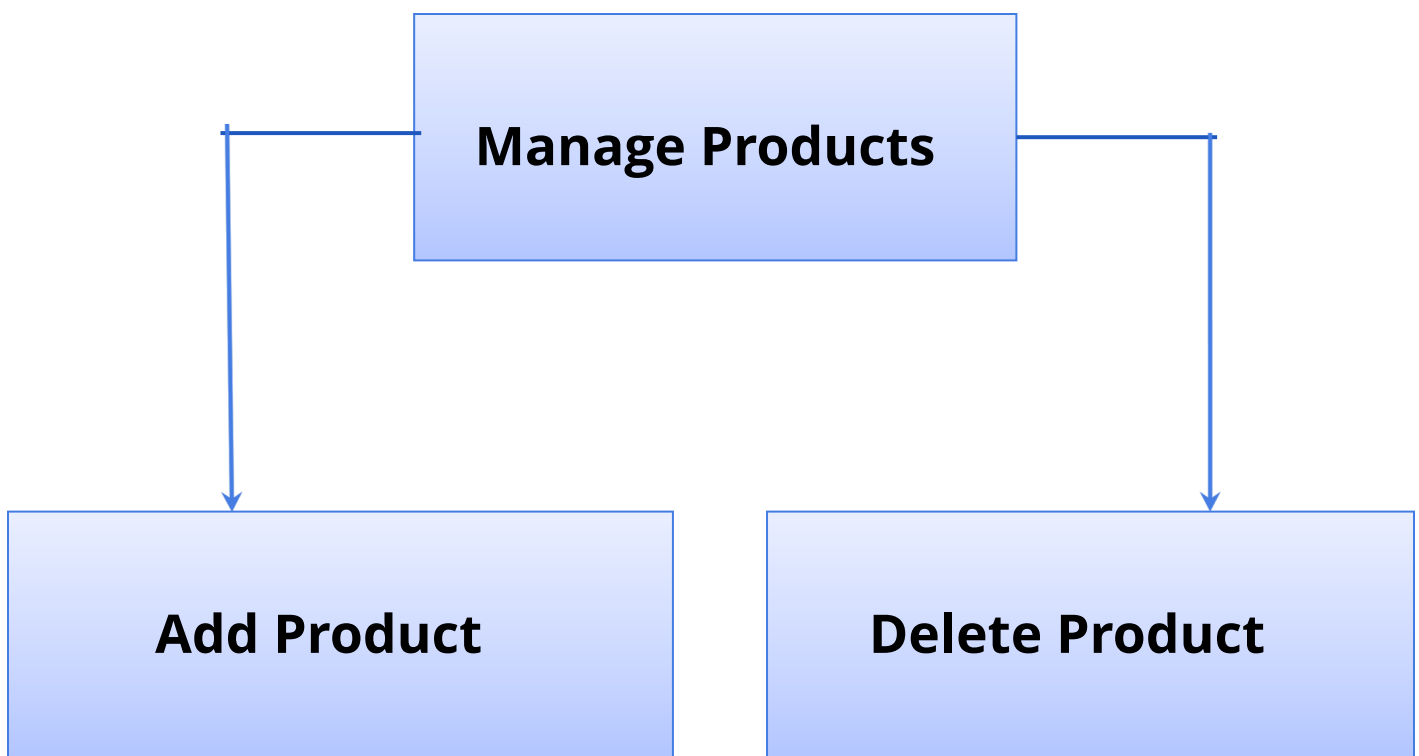
Users frequently forget items to buy because of which they have to run to shops again and again which is quite a frustrating and tiring situation and if our expenses crosses out budget while shopping that could be a matter of concern.

### **2.2 Proposed Solution**

To overcome this problematic situation I built a grocery app which helps you to list down all the item that you need to buy along with its price.

### 3. Theoretical Analysis

#### 3.1 Block Diagram



#### 3.2 Hardware/Software designing

- Windows 10 OS
- Android Studio

## 4. Experimental Investigations

In this project MVVM (Model View ViewModel) was used for architectural patterns, Room for database, Coroutines and RecyclerView to display the list of items.

**LiveData:** A data holder class that can be observed. Always holds/caches the latest version of data, and notifies its observers when data has changed. LiveData is lifecycle aware. UI components just observe relevant data and don't stop or resume observation. LiveData automatically manages all of this since it's aware of the relevant lifecycle status changes while observing.

**ViewModel:** Acts as a communication center between the Repository (data) and the UI. The UI no longer needs to worry about the origin of the data. ViewModel instances survive Activity/Fragment recreation.

**Repository:** A class that you create that is primarily used to manage multiple data sources.

**Entity:** Annotated class that describes a database table when working with Room.

**Room database:** Simplifies database work and serves as an access point to the underlying SQLite database (hides SQLiteOpenHelper). The Room database uses the DAO to issue queries to the SQLite database.

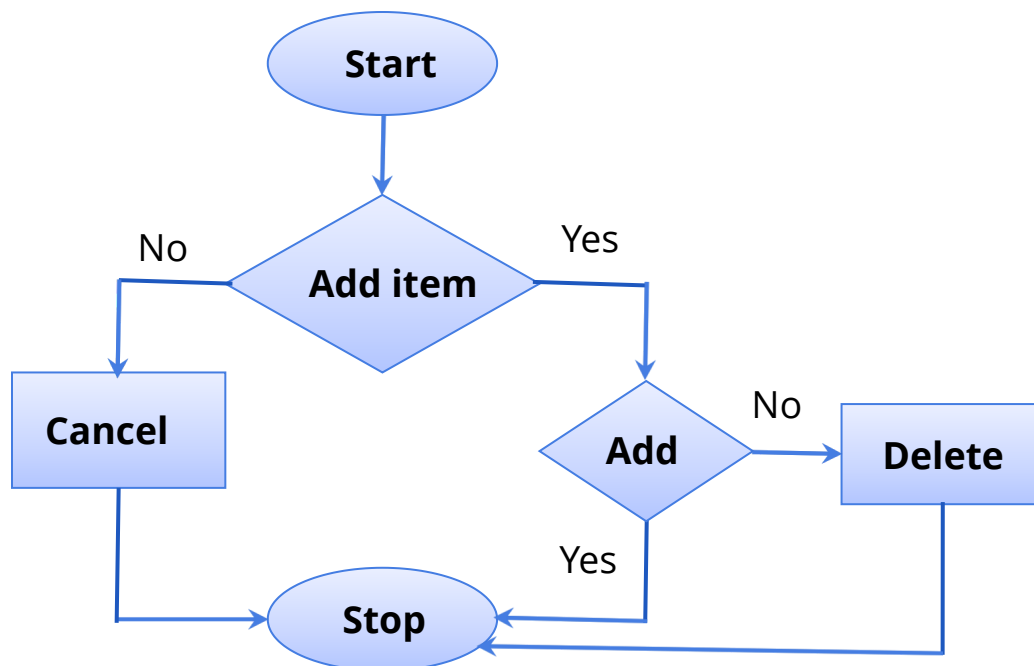
**SQLite database:** On device storage. The Room persistence library creates and maintains this database for you.

**DAO:** Data access object. A mapping of SQL queries to functions. When you use a DAO, you call the methods, and Room takes care of the rest.

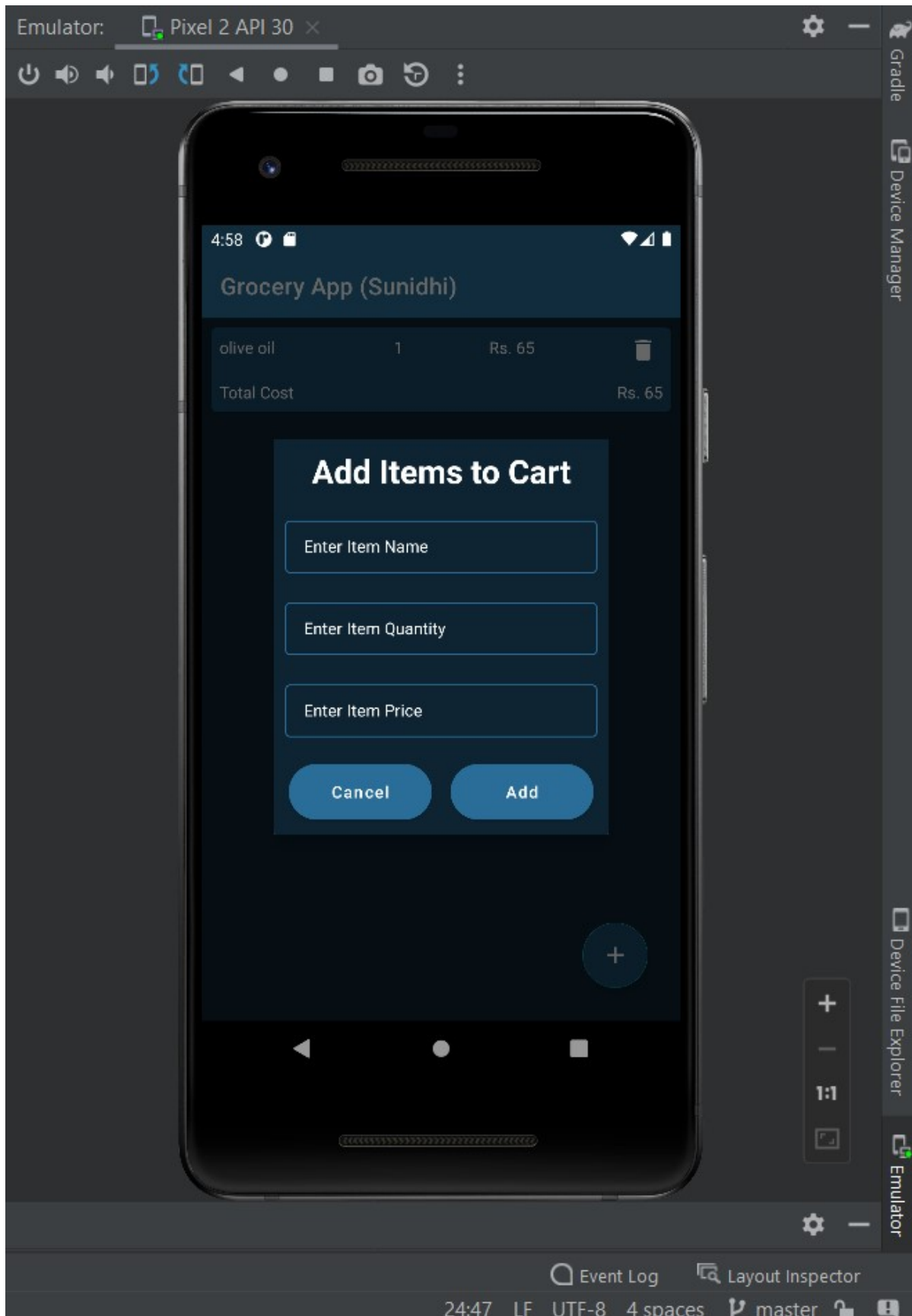
**RecyclerView:** It is a container and is used to display the collection of data in a large amount of dataset that can be scrolled very effectively by maintaining a limited number of views.

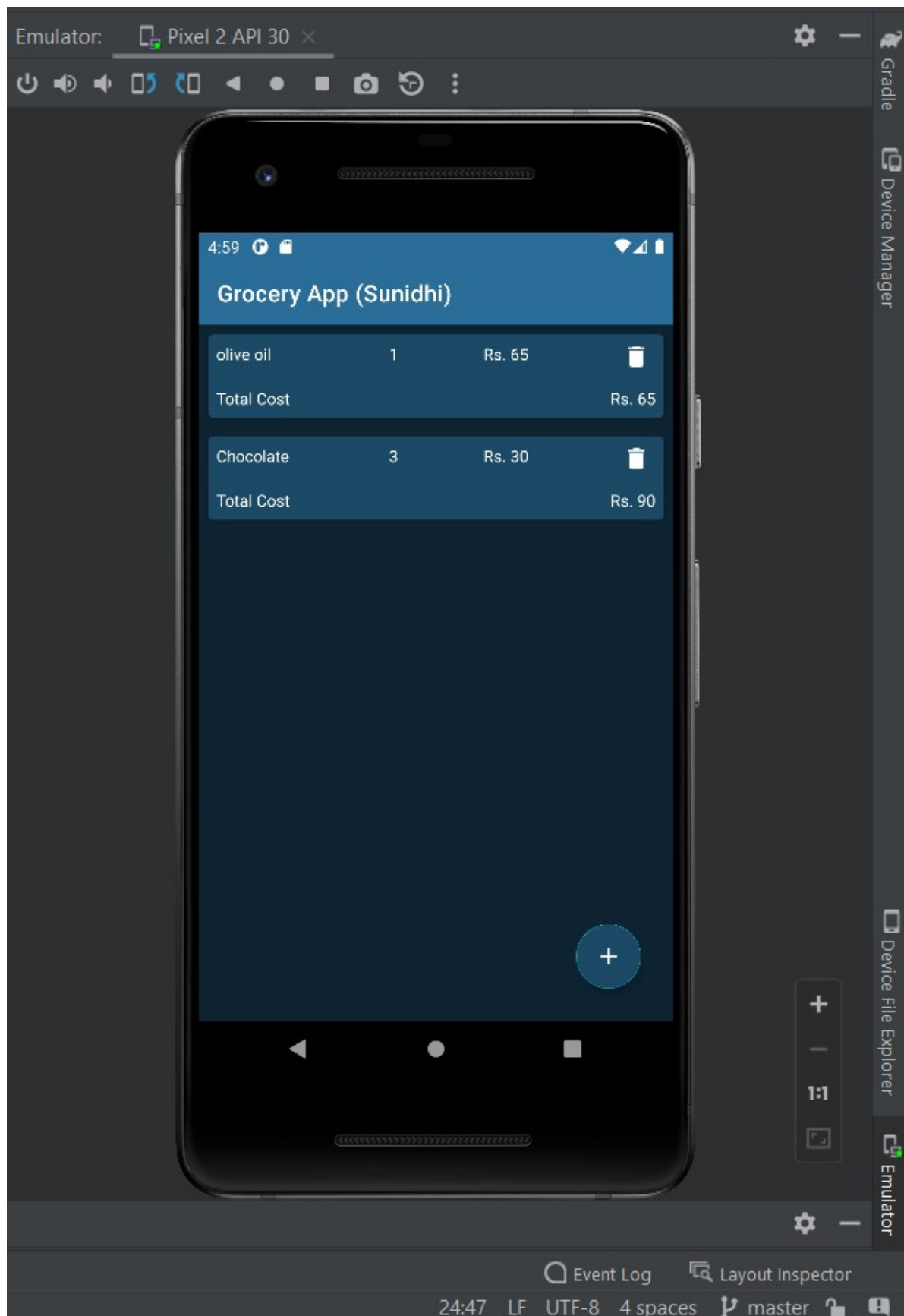
**Coroutines:** Coroutines are lightweight thread, we use a coroutine to perform an operation on other threads, by this our main thread doesn't block and our app doesn't crash.

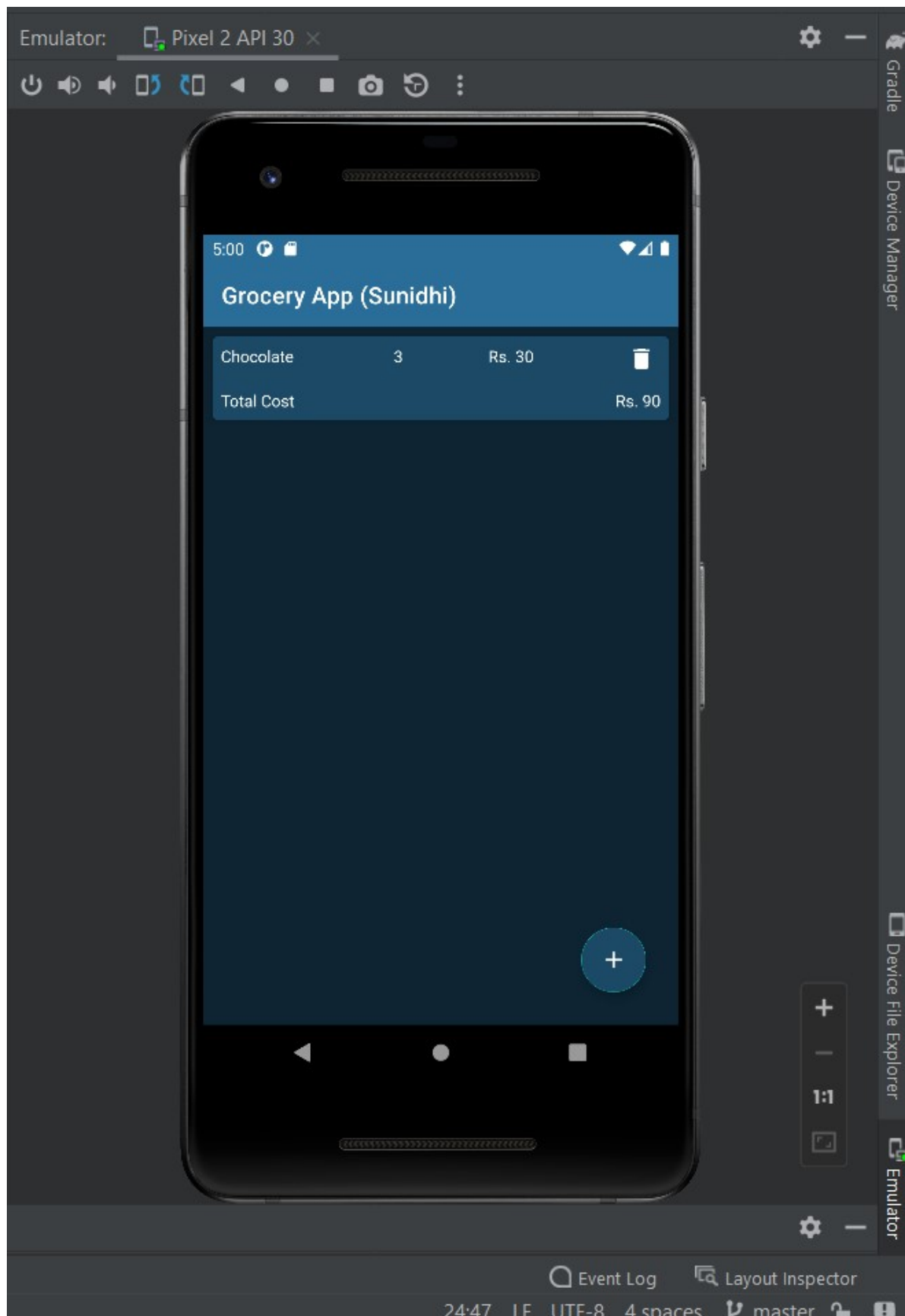
## 5. Flowchart



## 6. Result







## 7. Conclusion

This project helped me to clear my concepts on Room Database, Coroutines, MVVM, etc. This project would help me not just as a developer to learn new and interesting things but also as a user we generally forgets items to purchase while shopping. Working on this project made me confident enough to apply my knowledge on android app development and create such an app. I have used Kotlin to build this application. All the functionality is coded in the classes and interfaces created and the layout is designed using xml.

## 8. Reference

- **Google:** <https://www.google.com/>
- **Geeksforgeeks:** <https://www.geeksforgeeks.org/how-to-build-a-grocery-android-app-using-mvvm-and-room-database/>
- **Android Developer:** <https://developer.android.com/codelabs/android-room-with-a-view-kotlin#0>
- **YouTube:** [https://www.youtube.com/watch?v=vdcLb\\_Y71lc](https://www.youtube.com/watch?v=vdcLb_Y71lc)
- **SmartInternz:** [https://smartinternz.com/Student/guided\\_project\\_workspace/56568](https://smartinternz.com/Student/guided_project_workspace/56568)

## 9. Appendix

### 9.1 Source Code

MainActivity.kt

```
package com.example.groceryappsunidhi

import android.app.Dialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
```



```
import androidx.lifecycle.Observer
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.floatingactionbutton.FloatingActionButton

class MainActivity : AppCompatActivity(),
GroceryRVAdapter.GroceryItemClickInterface {
    lateinit var itemsRV: RecyclerView
    lateinit var addFAB: FloatingActionButton
    lateinit var list: List<GroceryItems>
    lateinit var groceryRVAdapter: GroceryRVAdapter
    lateinit var groceryViewModal: GroceryViewModel

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        itemsRV = findViewById(R.id.idRVItems)
        addFAB = findViewById(R.id.idFABAdd)
        list = ArrayList<GroceryItems>()
        groceryRVAdapter = GroceryRVAdapter(list, this)
        itemsRV.layoutManager = LinearLayoutManager(this)
        itemsRV.adapter = groceryRVAdapter
        val groceryRepository = GroceryRepository(GroceryDatabase(this))
        val factory = GroceryViewModelFactory(groceryRepository)
        groceryViewModal = ViewModelProvider(this,
factory).get(GroceryViewModel::class.java)
        groceryViewModal.getAllGroceryItems().observe(this, Observer {
            groceryRVAdapter.list = it
            groceryRVAdapter.notifyDataSetChanged()
        })

        addFAB.setOnClickListener {
            openDialog()
        }
    }

    fun openDialog() {
        val dialog = Dialog(this)
        dialog.setContentView(R.layout.grocery_add_dialogue)
        val btnCancel = dialog.findViewById<Button>(R.id.idBtnCancel)
        val addBtn = dialog.findViewById<Button>(R.id.idBtnAdd)
        val itemEdt = dialog.findViewById<EditText>(R.id.idEdtItemName)
```

```
        val itemPriceEdt = dialog.findViewById<EditText>(R.id.idEdtItemPrice)
        val itemQuantityEdt =
dialog.findViewById<EditText>(R.id.idEdtItemQuantity)
        cancelBtn.setOnClickListener {
            dialog.dismiss()
        }
        addBtn.setOnClickListener {
            val itemName: String = itemEdt.text.toString()
            val itemPrice: String = itemPriceEdt.text.toString()
            val itemQuantity: String = itemQuantityEdt.text.toString()
            val qty: Int = itemQuantity.toInt()
            val pr: Int = itemPrice.toInt()
            if (itemName.isNotEmpty() && itemPrice.isNotEmpty() &&
itemQuantity.isNotEmpty()) {
                val items = GroceryItems(itemName, qty, pr)
                groceryViewModal.insert(items)
                Toast.makeText(applicationContext, "Item Inserted..",
Toast.LENGTH_SHORT).show()
                groceryRVAdapter.notifyDataSetChanged()
                dialog.dismiss()
            } else {
                Toast.makeText(
                    applicationContext,
                    "Please Enter all the data..",
                    Toast.LENGTH_SHORT
                ).show()
            }
        }
    }
    dialog.show()
}

override fun onItemClick(groceryItems: GroceryItems) {
    groceryViewModal.delete(groceryItems)
    groceryRVAdapter.notifyDataSetChanged()
    Toast.makeText(applicationContext, "Item Deleted..",
Toast.LENGTH_SHORT).show()
}
}
```

activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/blue_shade_1"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/idRVItems"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/grocery_rv_item" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/idFABAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_margin="28dp"
        android:backgroundTint="@color/blue_shade_2"
        android:src="@drawable/ic_add"
        app:tint="@color/white" />

</RelativeLayout>
```

## GroceryDao.kt

```
package com.example.groceryappsunidhi

import androidx.lifecycle.LiveData
import androidx.room.*

@Dao
interface GroceryDao {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
suspend fun insert(item: GroceryItems)

@Delete
suspend fun delete(item: GroceryItems)

@Query("SELECT * FROM grocery_items")
fun getAllGroceryItems(): LiveData<List<GroceryItems>>

}
```

## GroceryItems.kt

```
package com.example.groceryappsunidhi

import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "grocery_items")
data class GroceryItems (

    @ColumnInfo(name = "itemName")
    var itemName: String,

    @ColumnInfo(name = "itemQuantity")
    var itemQuantity: Int,

    @ColumnInfo(name = "itemPrice")
    var itemPrice: Int,

)
{
    @PrimaryKey(autoGenerate = true)
    var id: Int? = null
}
```

## GroceryRepository.kt

```
package com.example.groceryappsunidhi
```

```
class GroceryRepository (private val db: GroceryDatabase) {  
  
    suspend fun insert(items: GroceryItems) = db.getGroceryDao().insert(items)  
    suspend fun delete(items: GroceryItems) = db.getGroceryDao().delete(items)  
  
    fun getAllItems() = db.getGroceryDao().getAllGroceryItems()  
}
```

## GroceryDatabase.kt

```
package com.example.groceryappsunidhi  
  
import android.content.Context  
import androidx.room.Database  
import androidx.room.Room  
import androidx.room.RoomDatabase  
  
@Database(entities = [GroceryItems::class], version = 1)  
abstract class GroceryDatabase : RoomDatabase() {  
  
    abstract fun getGroceryDao(): GroceryDao  
  
    companion object {  
        @Volatile  
        private var instance: GroceryDatabase? = null  
        private val LOCK = Any()  
  
        operator fun invoke(context: Context) = instance ?: synchronized(LOCK)  
{  
            instance ?: createDatabase(context).also {  
                instance = it  
            }  
        }  
  
        private fun createDatabase(context: Context) =  
            Room.databaseBuilder(  
                context.applicationContext,  
                GroceryDatabase::class.java,  
                "Grocery.db"  
            ).build()  
    }  
}
```

```
}
```

## GroceryRVAdapter.kt

```
package com.example.groceryappsunidhi

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class GroceryRVAdapter(
    var list: List<GroceryItems>,
    val groceryItemClickInterface: GroceryItemClickInterface
) : RecyclerView.Adapter<GroceryRVAdapter.GroceryViewHolder>() {

    inner class GroceryViewHolder(itemView: View) :
        RecyclerView.ViewHolder(itemView) {
        val nameTV = itemView.findViewById<TextView>(R.id.idTVItemName)
        val quantityTV = itemView.findViewById<TextView>(R.id.idTVQuantity)
        val rateTV = itemView.findViewById<TextView>(R.id.idTVRate)
        val amountTV = itemView.findViewById<TextView>(R.id.idTVTotalAmt)
        val deleteTV = itemView.findViewById<ImageView>(R.id.idTVDelete)
    }

    interface GroceryItemClickInterface {
        fun onItemClick(groceryItems: GroceryItems)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
        GroceryViewHolder {
        val view =
            LayoutInflater.from(parent.context).inflate(R.layout.grocery_rv_item,
                parent, false)
        return GroceryViewHolder(view)
    }

    override fun onBindViewHolder(holder: GroceryViewHolder, position: Int) {
        holder.nameTV.text = list.get(position).itemName
        holder.quantityTV.text = list.get(position).itemQuantity.toString()
    }
}
```

```
        holder.rateTV.text = "Rs. " + list.get(position).itemPrice.toString()
        val itemTotal: Int = list.get(position).itemPrice *
list.get(position).itemQuantity
        holder.amountTV.text = "Rs. " + itemTotal.toString()
        holder.deleteTV.setOnClickListener {
            groceryItemClickInterface.onItemClick(list.get(position))
        }
    }

    override fun getItemCount(): Int {
        return list.size
    }
}
```

strings.xml

```
<resources>
    <string name="app_name">Grocery App (Sunidhi)</string>
    <string name="quantity">Quantity</string>
    <string name="item_name">Item Name</string>
    <string name="rate">Rate</string>
    <string name="enter_item_name">Enter Item Name</string>
    <string name="enter_item_quantity">Enter Item Quantity</string>
</resources>
```

**Note :-** Since the page limit is exceeding I can't put the whole source code here. Please check the drive link or the github link below for full code.

**Drive link:**

<https://drive.google.com/drive/folders/1alrR8813ayQsUvbiHCpK2UwG7PRJrNGq>

**Github link:** <https://github.com/smartinternz02/Sl-GuidedProject-56568-1663733298>