**Grocery App Report**

**Introduction :**

**1.1Overview :**

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

**1.2Purpose :**

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**

https://martpluss.com/wp-content/uploads/2022/09/WhatsApp-Image-2022-09-25-at-9.18.39-PM.jpeg

**3.2 Hardware/Software designing**

● Windows 10 OS

1. An Android device or emulator

● Android Studio Manage Products Add Product Delete Product

**4 Experimental Investigation:**

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 latet version of data, and notifies its observers when data has changed. LiveData is lifec-ycle 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:**

[https://martpluss.com/wp-](https://martpluss.com/wp-content/uploads/2022/09/WhatsApp-Image-2022-09-25-at-9.18.48-PM-1.jpeg)content/uploads/2022/09/WhatsApp-Image-2022-09-25-at-9.18.48-PM-1.jpeg

6 Result:

## Adding Grocery to the Application

## [https://user-](https://user-images.githubusercontent.com/84894767/192084857-09844bfc-2a55-4fb5-a5ca-8de455f94ee1.jpeg)images.githubusercontent.com/84894767/192084857-09844bfc-2a55-4fb5-a5ca-8de455f94ee1.jpeg

## [https://user-](https://user-images.githubusercontent.com/84894767/192084899-d5a35e9d-e694-4ef7-a36e-c743eca0cfeb.jpeg)images.githubusercontent.com/84894767/192084899-d5a35e9d-e694-4ef7-a36e-c743eca0cfeb.jpeg

## 2. Grocery got added

## <https://user-images.githubusercontent.com/84894767/192084942-7f329066-56c7-41e0-a2a1-26a7ffedb754.jpeg>

## https://user-images.githubusercontent.com/84894767/192085062-27f85bf2-acf3-4538-9f8f-1382757dc4f4.jpeg

## 3) Item got deleted

## **<https://user-images.githubusercontent.com/84894767/192085091-f832aca6-1d2c-4366-9c38-d0c7b1c21e64.jpeg>**

**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 Codelabs: https://codelabs.developers.google.com/?cat=Android

● YouTube:<https://www.youtube.com/watch?v=vdcLb_Y71Ic>

● SmartInternz: https://smartinternz.com/

**9. Appendix:**

**9.1 Source Code:**

**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\_alignParentBottom="true"  
 android:layout\_margin="20dp"  
 android:backgroundTint="@color/blue\_shade\_2"  
 android:layout\_alignParentRight="true"  
 android:src="@drawable/ic\_add"  
 app:tint="@color/white" />  
</RelativeLayout>

**Grocery\_add\_dialog.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center">  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/blue\_shade\_1">  
 <TextView  
 android:id="@+id/idTVHeading"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="4dp"  
 android:gravity="center"  
 android:padding="4dp"  
 android:text="Add Item to Cart"  
 android:textAlignment="center"  
 android:textAllCaps="false"  
 android:textColor="@color/white"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/idTVHeading"  
 android:orientation="vertical">  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/TextInputEditTextTheme"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:hint="Enter Item name"  
 android:padding="5dp"  
 android:textColorHint="@color/white"  
 app:hintTextColor="@color/white">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/idEdtItemName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"  
 android:textColor="@color/white"  
 android:textColorHint="@color/white"  
 android:textSize="14sp" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/TextInputEditTextTheme"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:hint="Enter Item Quantity"  
 android:padding="5dp"  
 android:textColorHint="@color/white"  
 app:hintTextColor="@color/white">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/idEdtItemQuantity"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="number"  
 android:textColor="@color/white"  
 android:textColorHint="@color/white"  
 android:textSize="14sp" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 style="@style/TextInputEditTextTheme"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:hint="Enter Item price"  
 android:padding="5dp"  
 android:textColorHint="@color/white"  
 app:hintTextColor="@color/white">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/idEdtItemPrice"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="number"  
 android:textColor="@color/white"  
 android:textColorHint="@color/white"  
 android:textSize="14sp" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="5dp"  
 android:orientation="horizontal"  
 android:weightSum="2">  
  
 <Button  
 android:id="@+id/idBtnCancel"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="8dp"  
 android:layout\_weight="1"  
 android:background="@drawable/custom\_button\_back"  
 android:text="Cancel"  
 android:textAllCaps="false" />  
  
  
  
 <Button  
 android:id="@+id/idBtnAdd"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="8dp"  
 android:layout\_weight="1"  
 android:background="@drawable/custom\_button\_back"  
 android:text="Add"  
 android:textAllCaps="false" />  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 </RelativeLayout>  
  
</androidx.cardview.widget.CardView>

**Grocery\_rv\_item.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="8dp"  
 android:backgroundTint="@color/blue\_shade\_2"  
 app:cardCornerRadius="5dp"  
 app:cardElevation="4dp">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <LinearLayout  
 android:id="@+id/idLL1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:weightSum="5">  
  
 <TextView  
 android:id="@+id/idTVItemName"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="3dp"  
 android:layout\_weight="2"  
 android:padding="4dp"  
 android:text="Item Name"  
 android:textColor="@color/white" />  
  
 <TextView  
 android:id="@+id/idTVQuantity"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="3dp"  
 android:layout\_weight="1"  
 android:padding="4dp"  
 android:text="Quantity"  
 android:textColor="@color/white" />  
  
 <TextView  
 android:id="@+id/idTVRate"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="3dp"  
 android:layout\_weight="1.5"  
 android:padding="4dp"  
 android:text="Rate"  
 android:textColor="@color/white" />  
  
 <ImageView  
  
 android:layout\_width="0dp"  
  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="0.5"  
  
 android:layout\_margin="3dp"  
  
 android:padding="4dp"  
  
 app:tint="@color/white"  
  
 android:src="@drawable/ic\_delete"  
  
 android:id="@+id/idTVDelete"/>  
  
  
  
 </LinearLayout>  
  
  
  
 <TextView  
  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="wrap\_content"  
  
 android:id="@+id/idTVHeading"  
  
 android:layout\_below="@+id/idLL1"  
  
 android:layout\_margin="3dp"  
  
 android:layout\_toLeftOf="@+id/idTVToatalAmt"  
  
 android:padding="4dp"  
  
 android:text="Total Cost"  
  
 android:textColor="@color/white"/>  
  
  
  
 <TextView  
  
 android:layout\_width="wrap\_content"  
  
 android:layout\_height="wrap\_content"  
  
 android:id="@+id/idTVToatalAmt"  
  
 android:layout\_below="@+id/idLL1"  
  
 android:layout\_alignParentEnd="true"  
  
 android:layout\_alignParentRight="true"  
  
 android:layout\_margin="3dp"  
  
 android:padding="4dp"  
  
 android:text="Amt"  
  
 android:textColor="@color/white"/>  
  
 </RelativeLayout>  
  
  
  
</androidx.cardview.widget.CardView>

**MainActivity.kt**

package com.example.groceryapplication  
  
  
  
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: GroceryViewModal  
  
  
  
 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 = GroceryViewModalFactory(groceryRepository)  
  
 groceryViewModal = ViewModelProvider(this, factory).get(GroceryViewModal::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\_dialog)  
  
 val cancelBtn = 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()  
  
 }  
  
}

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.

Github link:<https://github.com/smartinternz02/SPSGP-79421-Virtual-Internship---Android-Application-Development-Using-Kotlin>