Smart Internz

Grocery App Report

1. Introduction

Overview

This is an androidapp that helps you to make a list ofgrocery items along with its price and quantity.

Purpose

We are humans and we cannot remember everything. Wesometimes 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 forbudget maintenance.

2.Literature Survey.

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.

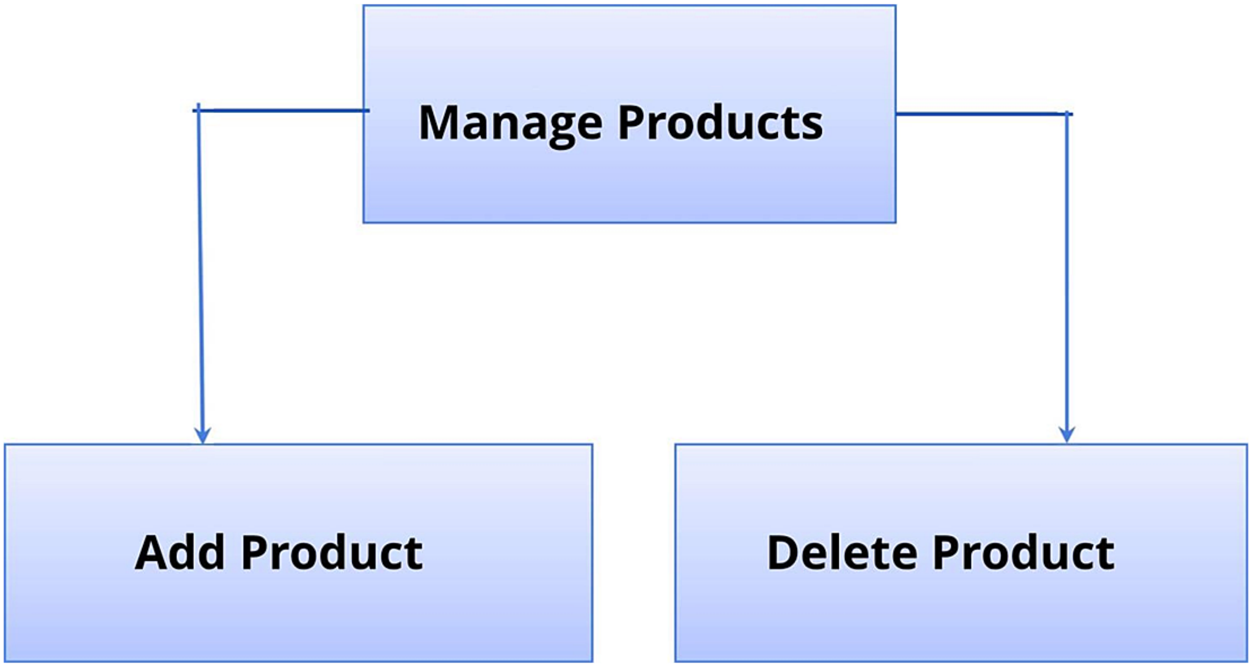
Proposed Solution

Smart Internz

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

3.Theoretical Analysis

a. Block Diagram



Hardware/Software designing

>windows 10 OS

>Andriod Studio

4.Experimental Investigations

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

Smart Internz

LiveData: A data holder class that can be observed. Always holds/caches the latest version of data, andnotifies 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 relevantlifecycle status changeswhile observing.

ViewModel: Acts as a communication center between the Repository (data)and the UI. The UI no longerneeds toworryabout the originof the data. ViewModel instancessurvive Activity/Fragment recreation.

Repository: A class that you createthat is primarilyusedto manage multiple data sources.

Entity: Annotatedclass that describes a database tablewhen 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.

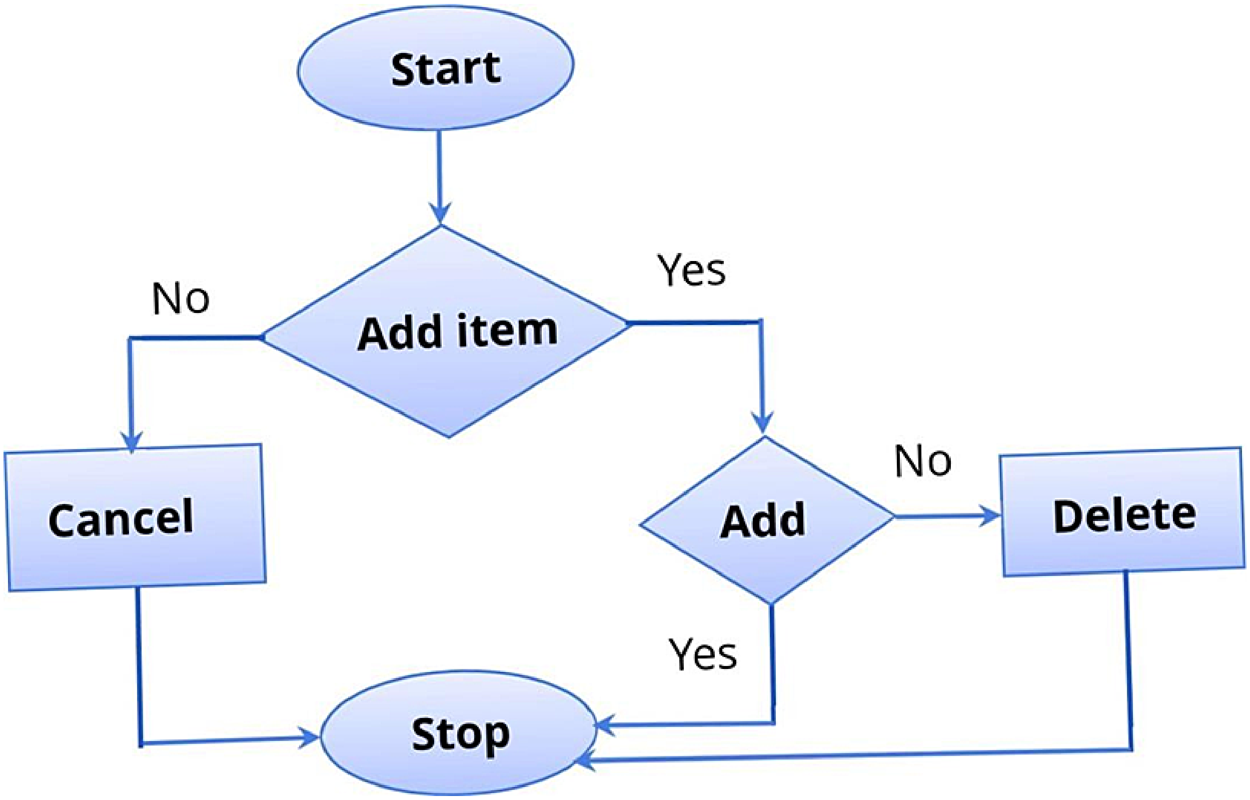
DAO: Data access object. A mapping of SQL queries to functions. When you use a DAO, you call the methods,andRoom takes care of the rest.

Smart Internz

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 numberof views.

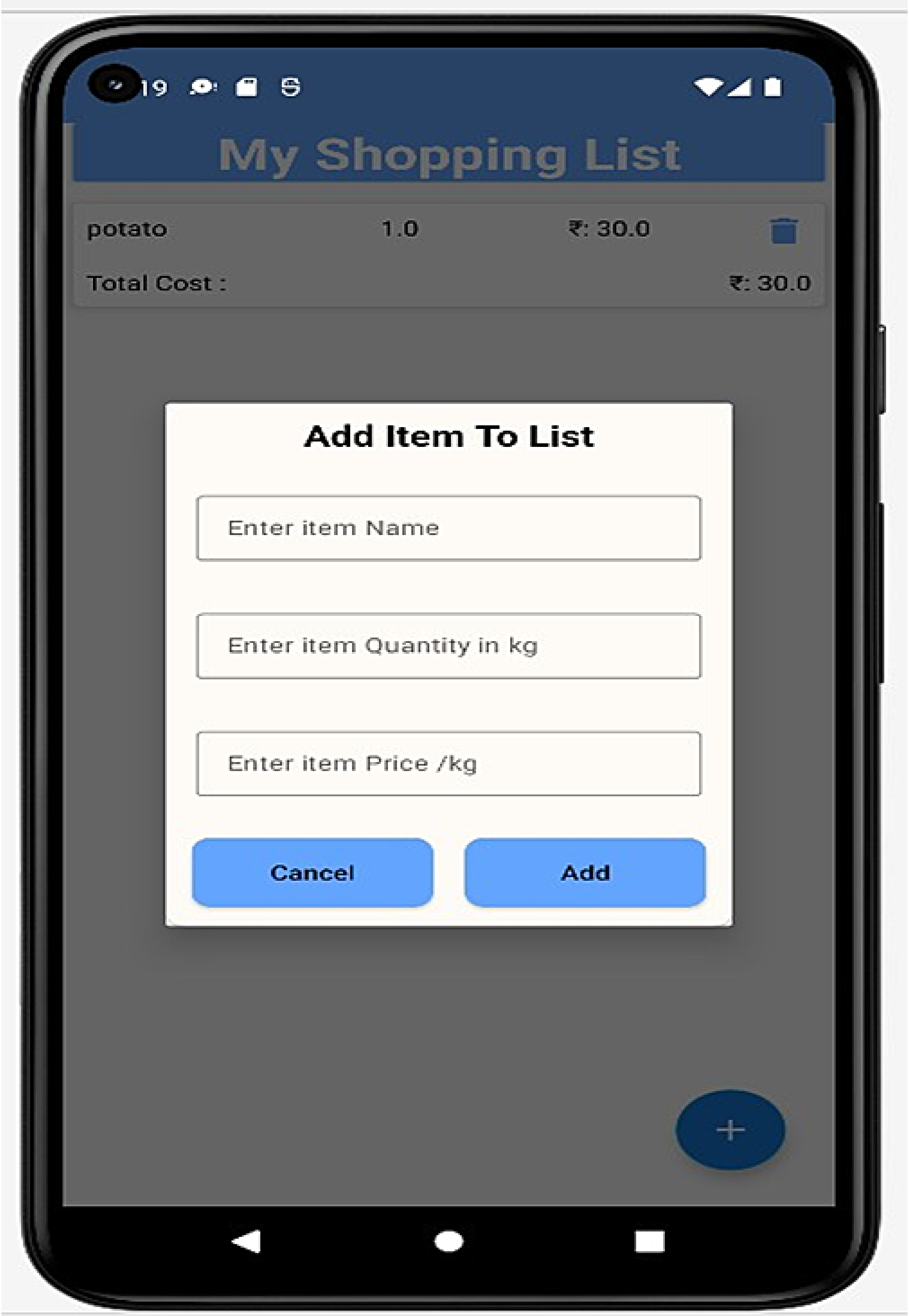
Coroutines: Coroutines are lightweight thread, we use a coroutine to perform an operation on other threads,by thisour main thread doesn't block and our app doesn'tcrash.

5.Flowchart



6.Result

Emulator : 2API 30

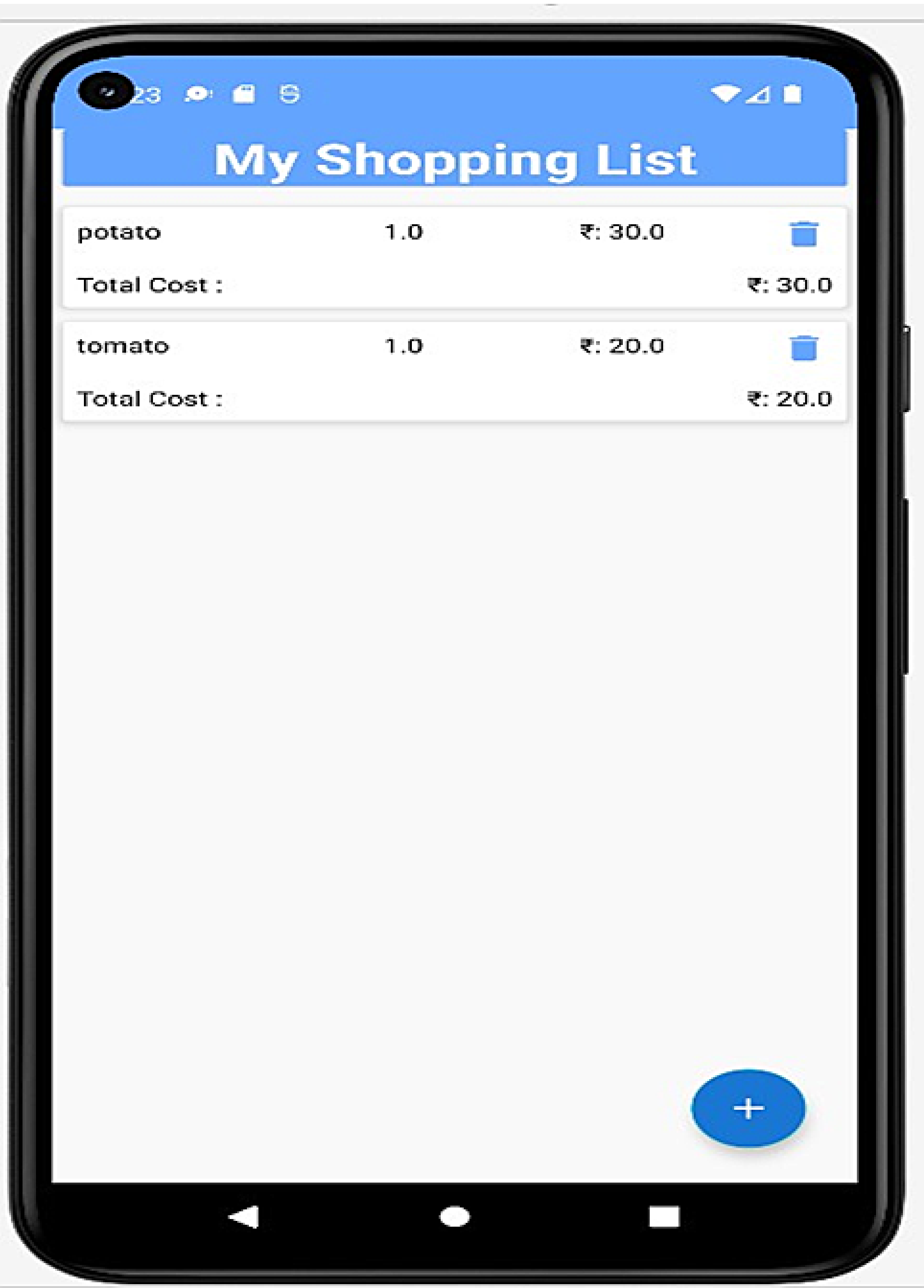


potato 1.0 ₹: 30.0

Total Cost : ₹: 30.0

Smart Internz

Emulator : 2API 30



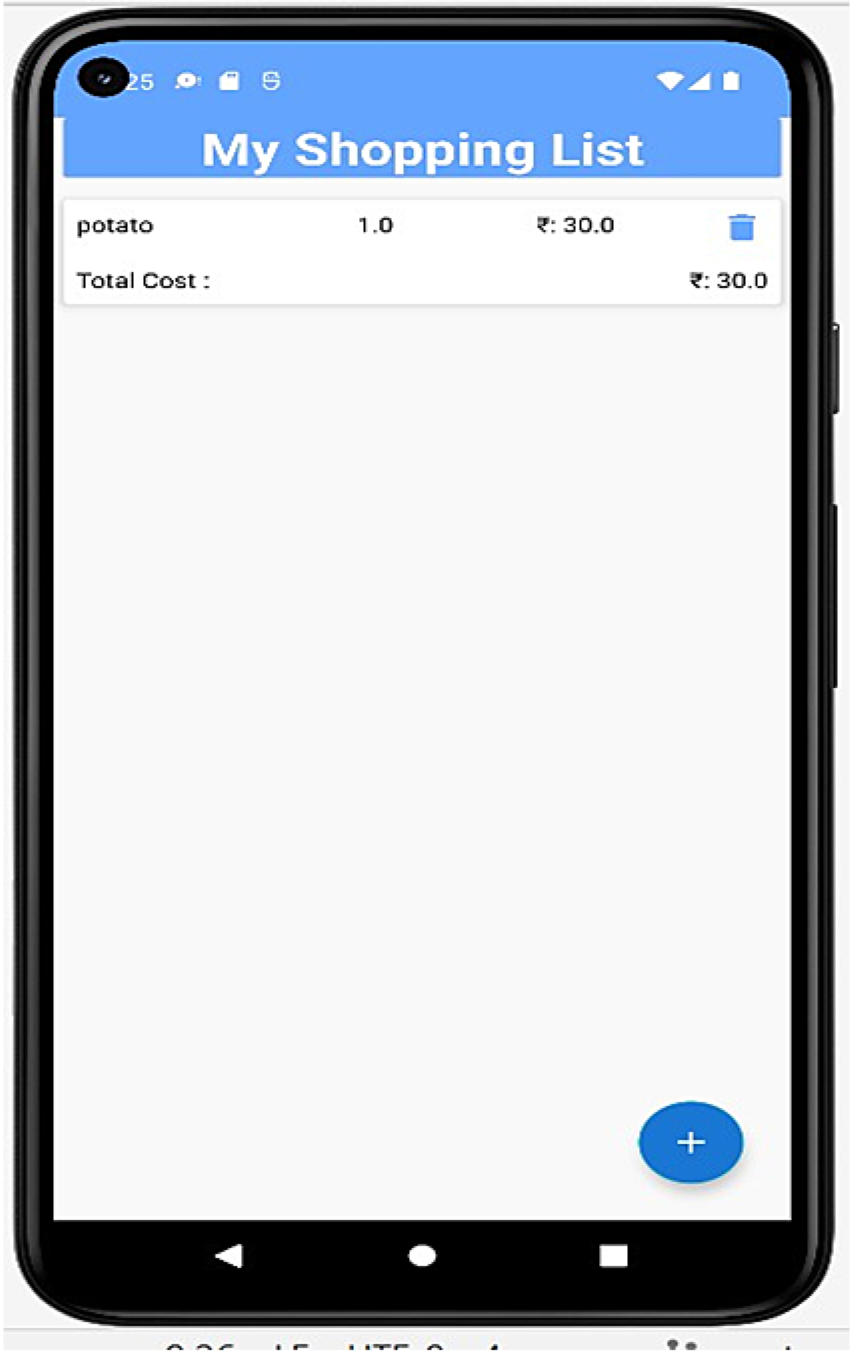
My Shopping List

potato	1.0	₹: 30.0	
Total Cost :			₹: 30.0
tomato	1.0	₹: 20.0	
Total Cost :			₹: 20.0

Event Log Layout inspector

Smart Internz

Emulator : 2API 30



My Shopping List

potato 1.0 ₹: 30.0

Total Cost : ₹: 30.0

Event Log Layout inspector

7.Conclusion

This project helped me to clearmy 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 codedin 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-androidapp-using-mvvm-and-room- database/

Android Developer:
hÜps://developer.android.com/codelabs/android-room- witha-view-kotlin#0

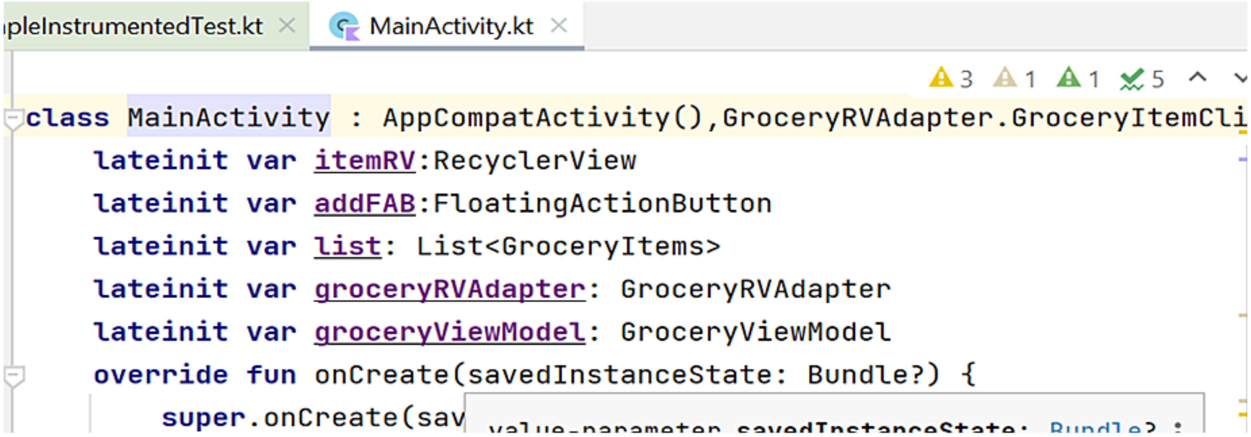
YouTube: hÜps://www.youtube.com/watch?v=vdcLb_Y71_lc

Smartlnternz:
hÜps://smartinternz.com/Student/guided_project_workspac e/55908

9.Appendix

a. Source Code

MainActivity.kt



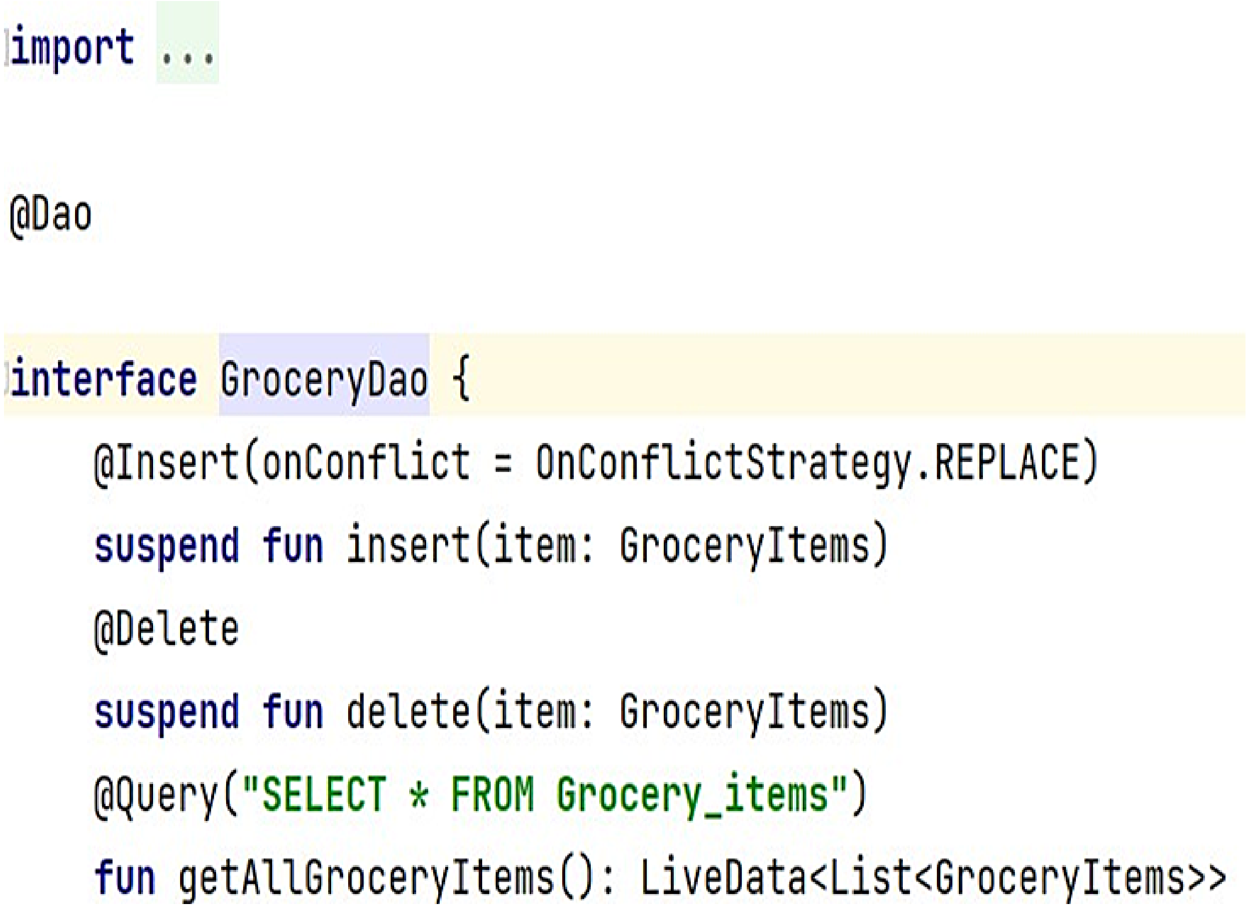
```
class MainActivity : AppCompatActivity(),GroceryRVAdapter.GroceryItemCli
    lateinit var itemRV:RecyclerView
    lateinit var addFAB:FloatingActionButton
    lateinit var list: List<GroceryItems>
    lateinit var groceryRVAdapter: GroceryRVAdapter
    lateinit var groceryViewModel: GroceryViewModel
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(sav
```

```
t11porL arid r Oidx.11 recycle . Observer

rnport androidx . li fecycle . ViewMode1Provider

import androidx . recyclerview . widget . LinearLayoutManager

import androidx . recyclerview . widget . RecyclerView

import com.google . android . material . float ingactionbutton .      on
       FloatingAct
•lass MainActivity : AppCompatActivity ( ) ,
GroceryRVAdapter . GroceryTtemClickTnterface {
    lateinit var itemsRV: RecyclerView
    lateinlt var adclFAB: FloatingActionButton
    lateinlt var list : List
    lateinlt var groceryRVAdapter:GroceryRVAdapter
    lateinit v  groceryViewModal: GroceryViewMode1
             ar
                onCreate      (savedInstanceState:
    override fun Bundle?)
        super    .    onCreate
        (savedInstanceState)setContentView
```

```
     (R. layout . acti vity_main)

    itemsRV = findViewById (R.id.
    idRVItems)addFAB =
    findViewByTd (R. id. idFABAdd)
    list -
    ( )
    graceryRVAdapter = GroceryRVAdapter (1 i st, this)
itemsRV  .  layoutManager  -  LinearLayoutManager  (this)
itemsRV.  adapter-  groceryRVAdapter  val  groceryRepository
- GroceryRepository  (GroceryDatabase  (this)  ) val  factory
=      GroceryViewMode1Factory      (groceryRepository)
groceryViewMoclal  =  ViewMode1Provider  (this,  factory)  .
get  (GroceryViewModeI  :  :class  .  java)  groceryvlewmodal  .
getA11GroceryIterns   (  )  .  observe  (this,  Observer  {
groceryRVAdapter.1ist      -      groceryRVAdapter      .
notifyDataSetChanged ( )
    addFAB   .   set
       OnC1ickListener
       openDia10g ( )




  fun   openDiaIog   (   )
    val  dialog  Dialog
    (this)   dialog   .
    setConter,tView
    (R. layout .
       cancelBtndialog .findViewById<Button> (R. id. idBtnCance1)
    val addBtn = dialog . (R. id. idBtnAdd)val itemEdt - dialog
    .   (R. id. idEdt Tternmame) dxalog .   (R.
    ce) val itemQuantityEdt = dialog .
    findViewById<EditText > (R . icl.idEdtItemQuanti
    ty) cancelBtn . setOnC1ickListener { dialog
    . dismiss ()

    addBtn . setOnC1ickListener val itemName: String
= itemEdt. text . tostring ( ) val itemPrice: String -
itemPriceEdt  .  text  .  toString  (  )  val  itemQuantity:
String = itemQuantityEdt . text . toString ( ) val qty:
   Int = itemQuantity . tolnt ( ) v al pr: Int itemPrice .
   tornt  (  )  i  f  (itemName  . isNotEmpty()  S  S  itemPrice.
   isNotEmpty()  itemQuantity  . isNotEmpty() ) val  items  =
   Groceryltems   (itemName,   qty,   pr)   groceryViewModa1.
   insert (items)
                Toast . makeText (applicationCont ext,Item
   Inserted.    .    roast.LENGTH_SHORT)    .    show    (    )
   groceryRVAdapter .
                   notifyDataSetChanged
                   ( ) dialog.dismiss ()
 Toast . makeText (applicationCon text,'1 Please Enter all  data. .
'1 ,Toast . LENGTH-SHORT) . show ( )



 dialog . show ()



       override  fun  onltemC1ick  (groceryltems:
           GroceryItems)groceryV1ewModa1 . delete
           (groceryltems) groceryRVAdapter . not
           ifyDat aSetChanged ( ) Toast .
           makeText(applicationContext,Item
           Deleted. .

   roast.LENGTH-SHORT) . show ( )
```

Smart Internz

GroceryDao.kt

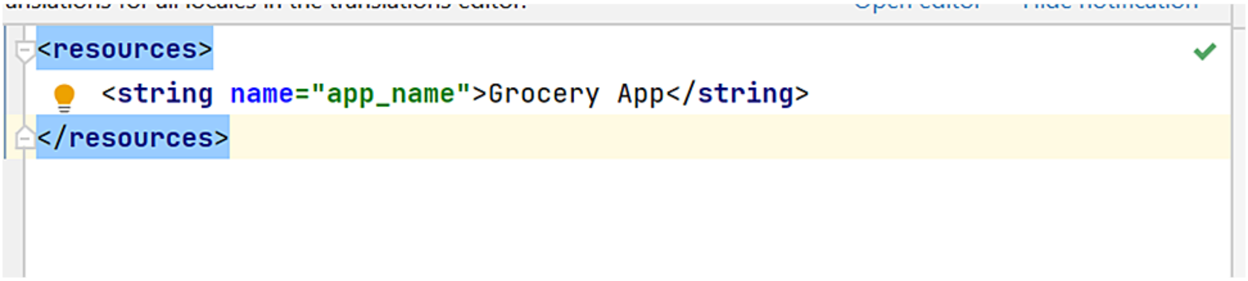


```
import ...


@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>>
```

```
GroceryRVAda

pter.ktPacka

ge

 com.example.groceryappsunidhi

 mport  android. view . Layout Inflater
import  android. view . View

import  android. view . ViewGroup

 mpo    android. widget . ImageView
 rt


 rnpo rt android. widget . TextView
 mport  androidx . recyclerview . widget .
```

```
       RecyclerView

--•lass  GroceryRVAdapter
    ( var list:
    val groceryItemC1ickInterface:
    GroceryItemC1ickInterface RecyclerView .
    Adapter<GroceryRVAdapter . GroceryViewE01der> ( )

    inner class GroceryViewH01der
(itemView: View)RecyclerView . ViewH01der
(itemView) (
        val nameTV - itemView.     (R.      id.iciTVItemlVame)
        val
        quantityTV - itemView .     (R. id.idTVQuantity)
        val rateTV= iternView.findViewByTd<TextView> (R.id. idTVRate)
        val amount TV - itemView .                  (R.             id.
                                                    idTVTota1Amt)
        val deleteTV = itemView .                    (R. id.
                                                     idTVDe1ete)
    interface
        GroceryItemC1ickInterface    {
        fun                onltemC1ick
        (grocerylterns               :
        GroceryItems)



    override fun onCreateViewH01der (parent: ViewGroup,
viewType : Int):GroceryViewHoIder      {
        val view
     Layout Inflater . from (parent . context) . inflate (R. layout
.    tern,oarent        ,            false) return
        GroceryViewH01der (view)



    override fun onBindViewH01der (holder: GroceryViewH01der,
        position:  Int)  holder  .  nameTV  .  text  =  list.  get
        (position) . itemName holder . i tv TV . text l i st .
        get (position) . i temQu:anti t.v
        .toStrinq()
```

strings.xml



```
<resources>
    <string name="app_name">Grocery App</string>
</resources>
```

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

Drive link:
https://drive.google.com/file/d/1JbBKwASgGicXqYgrzSk1Jbf4swBvvBpV/view?usp=sharing
Github link:
https://github.com/smartinternz02/SI-GuidedProject-54746-1661574219