GROCERY LIST APP
USING KOTLIN AND ANDROID STUDIO

SUBMITTED BY:

PRATHEEKSHA

GOOGLE SUPPORTED

Virtual Internship – Android Application Development Using Kotlin

Table Of Content

- The Areas Where This SolutionCan Be Applied

9.Conclusions

- Conclusion Summarizing the Entire Work and Findings.

10.Future Scope

- Enhancements That Can Be Made in The Future.

11.Bibliography

- References Of Previous Works or WebsitesVisited/Books Referred For

- Analysis About the Project,Solution Previous FindingsEtc.
- **Appendix**
- **A. SourceCode**

1.Introduction

1. **Overview:**

   Grocery lister app, android based application helps user to simplify the daily chaos of remembering the stuff they need to buy. This provides active database to list items out and its price in very efficient way. In this project,we are using mvvm (model view viewmodel) for architectural patterns, room for database, coroutines and recycler view to display the list of items.

2. **Purpose:**

   Grocery lister app aims to provide a comfortable experience while shopping daily need items. This project helps in assisting them in theirtasks by providing a user-friendly interface to create list of items they want to buy and keep track of their purchases. This helps them to simplify the calculations by totalling the amount also making it easier to help them remember by saving it in data base unless deleted. Its functionality also includes the taking note of items with quantity and amount mention.

2.Literature survey

a. Existing problem:

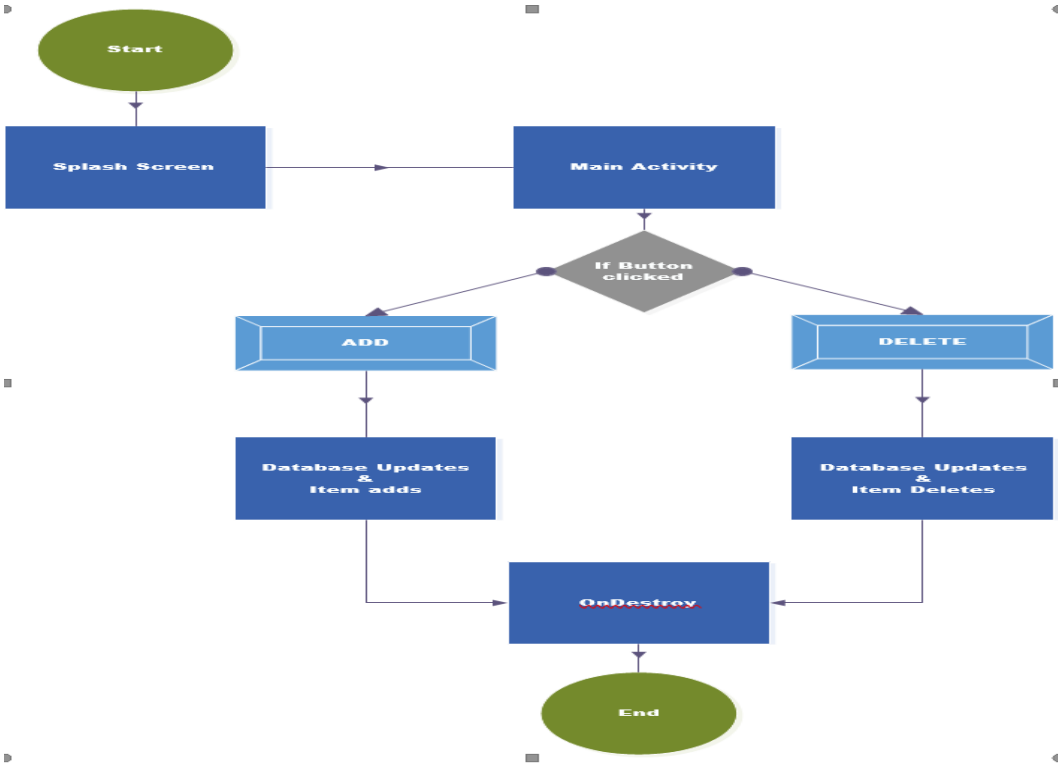
In general scenario most people go shopping while keeping notes in their head which they eventually end up forgetting about items they needed to buy, even while they keep it onpaper note and carry it all along it has been seen that notes can actually get detroit while travelling due to bad weather and also it gets difficult to add items while its on your mind and you in real crowdy market.To deal with all scenariosfor one and all there shall be something more reliable and durable whichhelps people to carry their list which can't be damaged easilyin addition to which also allows to add items far easily.

b. Proposed solution:

For the easiness of people and making human mind less occupied over tiny stuff but also giving a new ease way to make their list of items in digital and efficient way, the grocery lister, an app with functionalities which allows adding and deleting of items in user friendly way also helps them note down price and do total with quantities provided.

3.Theoretical analysis

a.Block diagram



On start of application the splash screen shows up which constitutes app icon, it will be shown for almost around 1.7 seconds. After the splash screen goes away main screen comes in notice consisting of two buttonsnamely add and delete. Add button by default is at bottom while delete button appears when item is already added.After all requiredoperations get done and app is been closed then after just action has been taken place it leads to updatingof database eitherdeletion or additionof itemsfrom it.

b.Hardware / Software designing:

Software:

The Software Package is developedusing Kotlin and Android Studio,basic SQL commandsare used to store the database.

Operating System:

Windows 11
Software Languages:Kotlin
andJava

Emulator:
Pixel 4 API 30 3.2

Hardware:

RAM: 16 GB RAMROM: 20
GB ROM

4.Theoretical analysis

- The projecthas been developed using Kotlin programming language.
- The software,Android Studio used to developapplication.
- For other testing purposesthe emulator has been used and also been testedupon Android phone (specific model: Samsung A8+).

- MVVM architecture in android is used to give structure to the project's code and understand code easily. MVVM is an architectural designpattern in android. MVVM treat Activity classes and XML files as View. This design pattern separates UI from its logic. Here is an image to quickly understand MVVM.
- Room Database persistence library is a database management library and it is used to store the data of apps like grocery item name, grocery item quantity, and grocery item price. Room is a cover layer on SQLite which helps to perform the operation on the database easily.
- RecyclerView is a container and it is used to display the collection of data in a large amount of data set that can be scrolled very effectively by maintaining a limited number of views.
- Coroutines are a 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.

Step By Step

ProcessStep 1: Create a New Project

To create a new projectin Android Studio please refer to Android Studio. Note that select **Kotlin** as the programming language.

Step 2: Before going to the coding section first you have to do some pre-task

Before going to the coding part first add these libraries in your gradle file and also apply the plugin as 'kotlin-kapt'. To add these library go to **Gradle Scripts > build.gradle (Module: app)**.

Step 3: ImplementRoom Database

a. Entities class

The entitiesclass contains all the columnsin the database and it should be annotated with @Entity (tablename = "Name of table"). Entity class is a data class. And @Column info annotation is used to enter column variable name and datatype.We will also add Primary Key for auto-increment. Go to **app > java > com.example.application-name.** Right-click on **com.example.application-name** go to new and create Kotlin file/class and name the file as **GroceryEntities**. See the code below to completely understand and implement.

b. DAO Interface

The DAO is an interface in which we create all the functionsthat we want to implementon the database. This interfacealso annotated with @Dao. Now we will create a function using suspend function which is a coroutines function.Here we create three functions,First is the insert function to insert items in the databaseand annotated with @Insert, Second is for deleting items from the database annotatedwith @Delete and Third is for getting all items annotated with @Query. Go to the **app > java > com.example.application-name**. Right- click on **com.example.application-name** go to new and create Kotlin file/class and name the file as **GroceryDao**. See the code below to implement.

c. Database class

Database class annotated with @Database(entities = [Name of Entity class.class], version = 1) these entitiesare the entities array list all the data entitiesassociating with the database and version shows the currentversion of the database. This database class inherits from the Room Database class. In **GroceryDatabase** classwe will make an abstract method to get an instance of DAO and further use this method from the DAO instanceto interact with the database. Go to the **app > java > com.example.application-name**. Right-clion **com.example.application-name** go to new and create Kotlin file/class as **GroceryDatabase**.

Step 4: Now we will implementthe Architectural Structurein the App

a.Repository class

The repositoryis one of the design structures. The repository class gives the data to the ViewModelclass and then the ViewModelclass uses that data for Views. The repository will choose the appropriate data locally or on the network. Here in our Grocery Repositoryclass data fetch locally from the Room database. We will add constructor value by creatingan instance of the databaseand stored in the db variable in the Grocery Repository class. Go to the **app > java > com.example.application-name**. Right-clic on **com.example.application-name** go to new and create Kotlin file/classas **GroceryRepository**. Go to **app > java > com.example.application-name**. Right-click on **com.example.application-name** go to new and create a new Package called**UI** and thenright-click on UI package and create a Kotlin file/class.

a.ViewModel class

ViewModel class used as an interfacebetween View and Data. Grocery View Model class inheritfrom View Model class and we will pass constructor value by creatinginstance variable of Repository class and stored in repositoryvariable. As we pass the constructor in View Model we have to create another class which is a Factory View Model class. Go to **app > java > com.example.application-name > UI**. Right-click on the UI package andcreate a Kotlin file/classand name the file as **GroceryViewModel**.

b.FactoryViewModel class

We will inherit the Grocery ViewModelFactory class from ViewModelProvider. NewInstanceFactory and again pass constructor value by creating instance variable of

GroceryRepository and return GroceryViewModel (repository). Go to the **app > java > com.example.application-name > UI**. Right-click on the **UI** package and create a Kotlin file/class name it **GroceryViewModelFactory**.

Step 5: Now let's jump into the UI part

In the **activity_main.xml** file, we will add two ImageView,RecyclerView, and Button after clicking this button a **DialogBox**open and in that dialog box user can enter the item name, itemquantity, and item price.

Step 6:

Let's implement **RecyclerView**. Now we will code the UI part of the row in the list. Go to **app> res > layout**. Right-click on layout, go to new, and then add a **Layout Resource File** and name it as **GroceryAdapter**. We will code adapter class for recyclerview. In the GroceryAdapter class, we will add constructor value by storingentities class as a list in list variable and create an instance of the view model. In Grocery Adapterwe will override three functions: onCreateViewHolder, getItemCount, aonbindViewHolder, we will also create an inner class called grocery view holder. Go to the **app > java > com.example.application- name**.Right-click on **com.example.application-name** go to new and create a new Package called **Adapter** and then right-click on Adapter package and create a Kotlin file/class nameit **GroceryAdapter**.

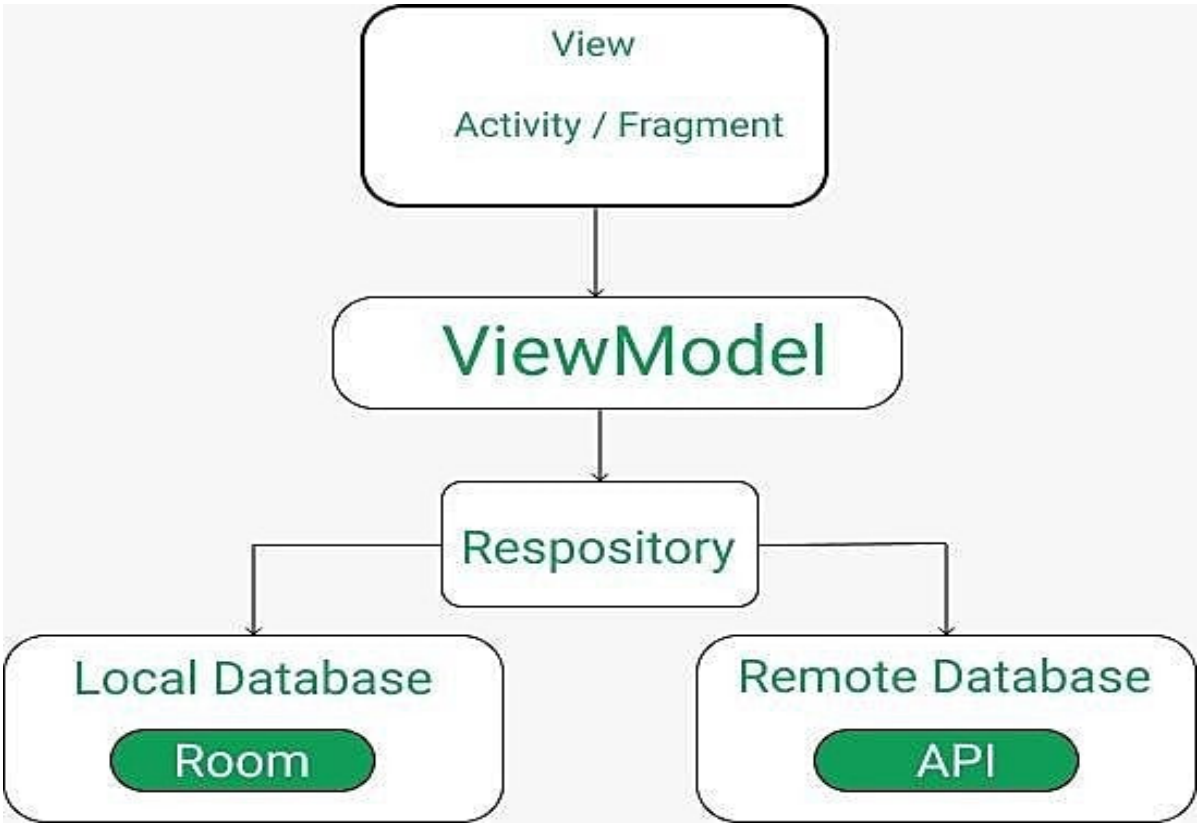
Step 7

To enter grocery item, quantity, and price from the user we have to create an interface.To implement this interface we will use DialogBox. First create UI of dialog box. In this dialog box we will add three edit text and two text view. Three edit text to enter grocery item name, quantityand price. Two text view one for save and other for cancel. After clicking the save text all data saved into the databaseand by clicking on the cancel text dialog box closes. Go to the **app > res > layout**. Right-click on **layout**, go to new and then add a **Layout Resource File** and name it as **GroceryDialog**. To add a clicklistener on save text we have to create an interface first in which we create a function.Go to the **app > java > com.example.application- name > UI.** Right-click on the **UI** package and create a Kotlin file/class and create an **interface** name it as **DialogListener**.

Step 8:

In this final step we will code in our **MainActivity**. In our **MainActivity**, we have to set up the recyclerview and add click listener on add button to open the dialog box.

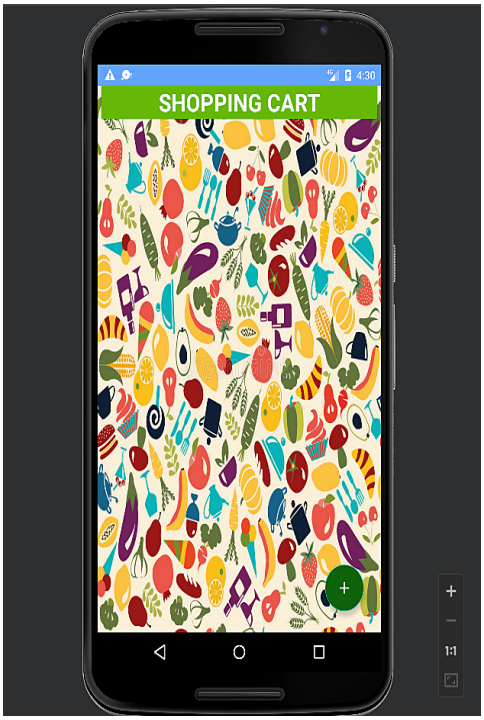
5.Flowchart



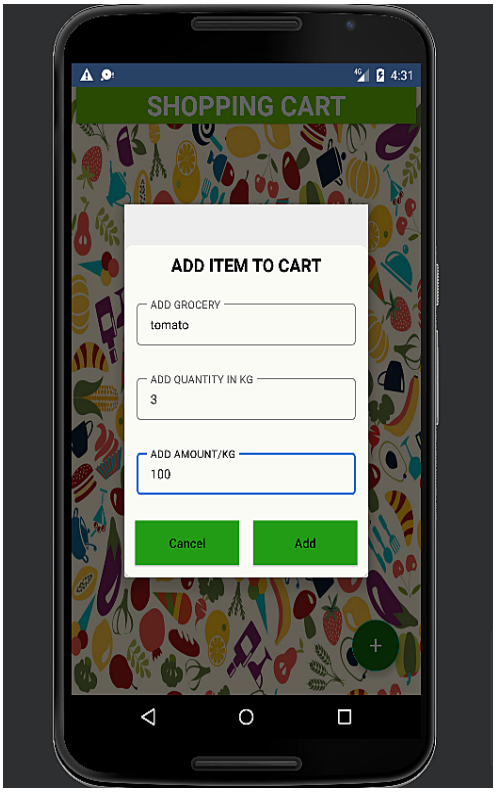
6.Result

Grocery Lister App, Android based application helps user to simplify the daily chaos of remembering the stuff they need to buy. This project helps in assisting them in their tasks by providing a user-friendly interface to create list of itemsthey want to buy and keep track of their purchases. This helps them to simplifythe calculations by totalling the amount also making it easier to help them remember by saving it in databaseunless deleted. Its functionality also includes the taking note of items with quantity and amount mention.

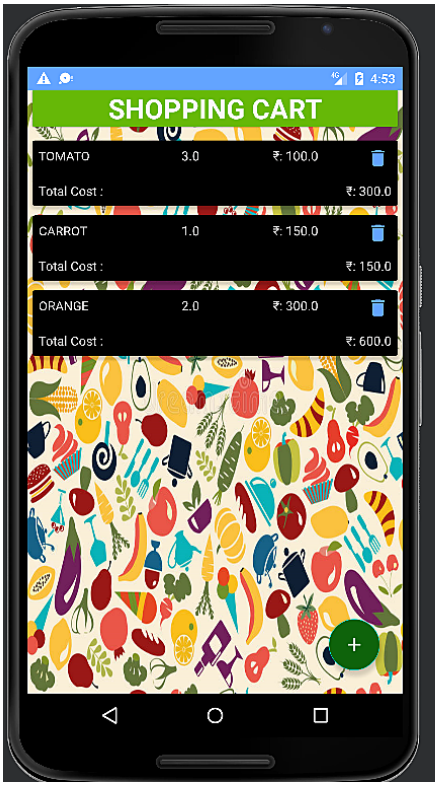
APP LOGO



MAIN PAGE OF APP



ADD AN ITEM DIAGLOGUE BOX



AFTER ADDING ITEM

7.Advantages &Disadvantages

Advantages:

1. Scope of this project includes giving users the ability to make lists of productsthey willing to purchase.
2. Application helps user to simplify the calculations by totallingthe amount whilealso making it easier to help them remember by saving it in databaseunless deleted.
3. Its functionality also includes the taking note of items with quantity and amount mention.

Disadvantages:

4. This project doesn't give users the ability to buy a product from a shop (online).
5. It doesn't include information from real bank account to create an expense history.

8.Applications

Its application is but not limited to making items listing, managing items and price of data or itemsin the list. It can further be used to manage a total financial evaluation by total pricing it. Further, it can be applied to transaction history and which helps to evaluate it too.

9.Conclusion

This project helps in assisting them in their tasks by providing a user- friendly interface to create list of items they want to buy and keep track of their purchases. This grocery application will help to store the list of data items include name of item, price and quantity required. Admins store his/her data in the list, the grocery application very helpful to users.

10.Future Scope

This application helps to store the list of items by Admin. But also can integrate the transaction management so one can actually evaluate their financial expenditure

In Future we can also add scheduledaddition of items according to requirement of user.

The Featuresare:

- Add User Panel
- Add AdminPanel
- Provide Login Authentication
- Add Image to user Product and Rating

11.**Bibliography**

https://www.geeksforgeeks.org/introduction-to-kotlin/

https://www.geeksforgeeks.org/introduction-to-android-development/#:~:text=Google%20first%20publicly%20announced%20Android,with%20the%20version%20Android%201.0

https://developer.android.com/courses/android-basics-kotlin/unit-1

https://developer.android.com/courses/android-basics-kotlin/unit-2

https://developer.android.com/courses/android-basics-kotlin/unit-3

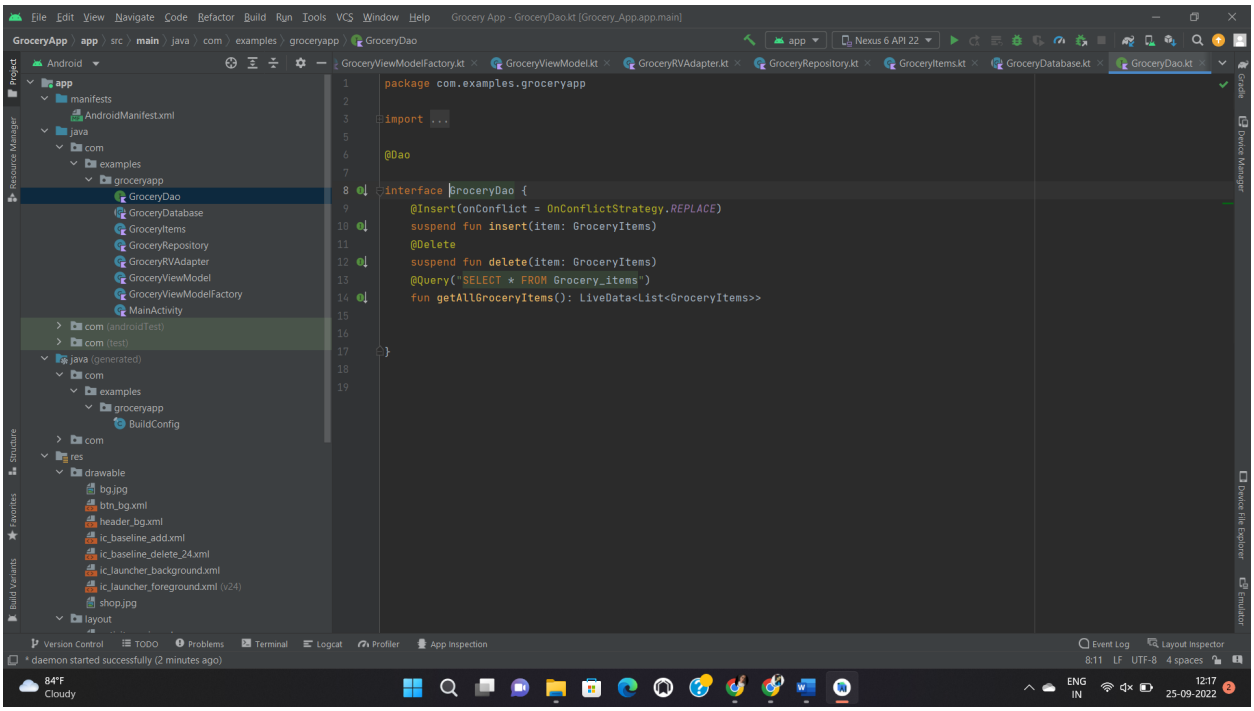
https://developer.android.com/courses/android-basics-kotlin/unit-4

https://developer.android.com/courses/android-basics-kotlin/unit-5

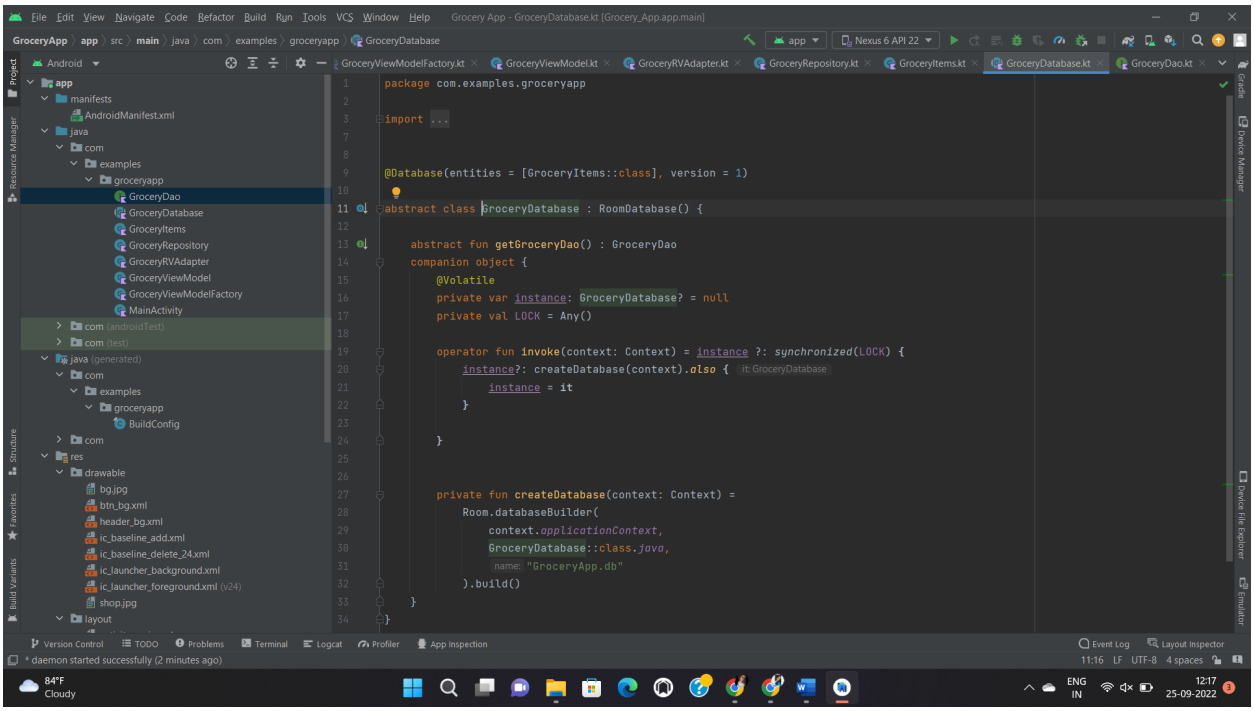
https://developer.android.com/courses/android-basics-kotlin/unit-6

APPENDIX

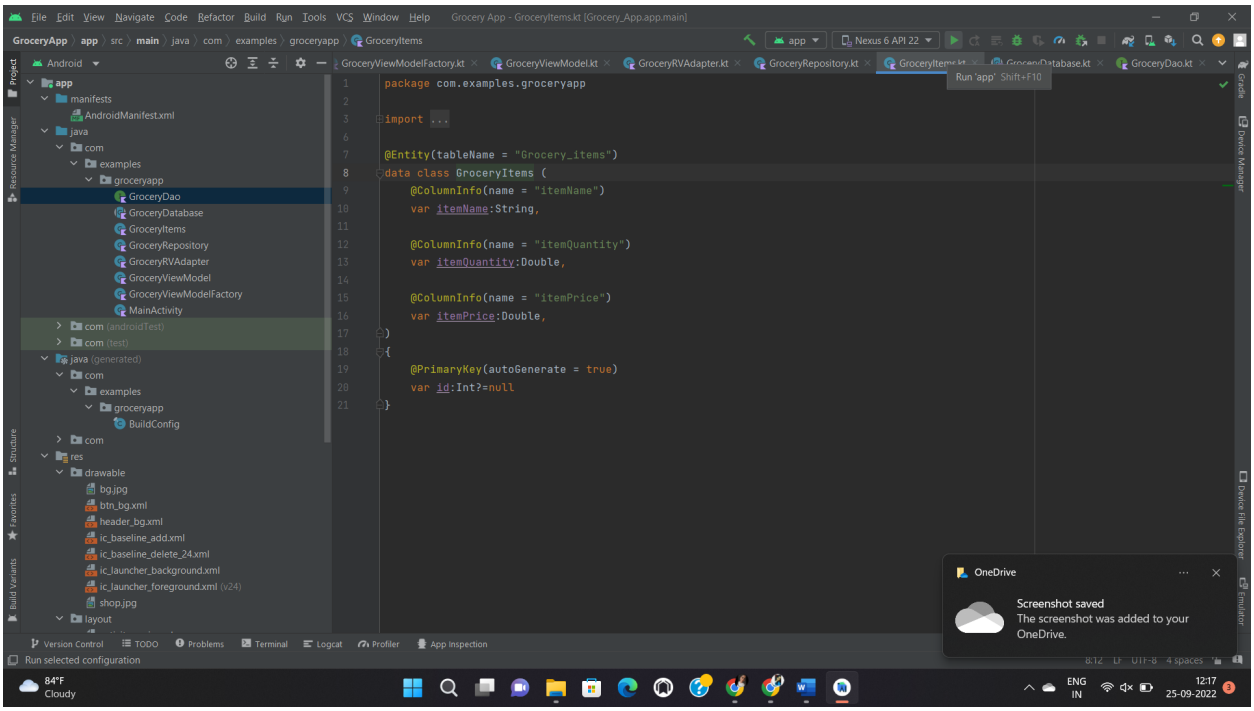
a. Source Code:

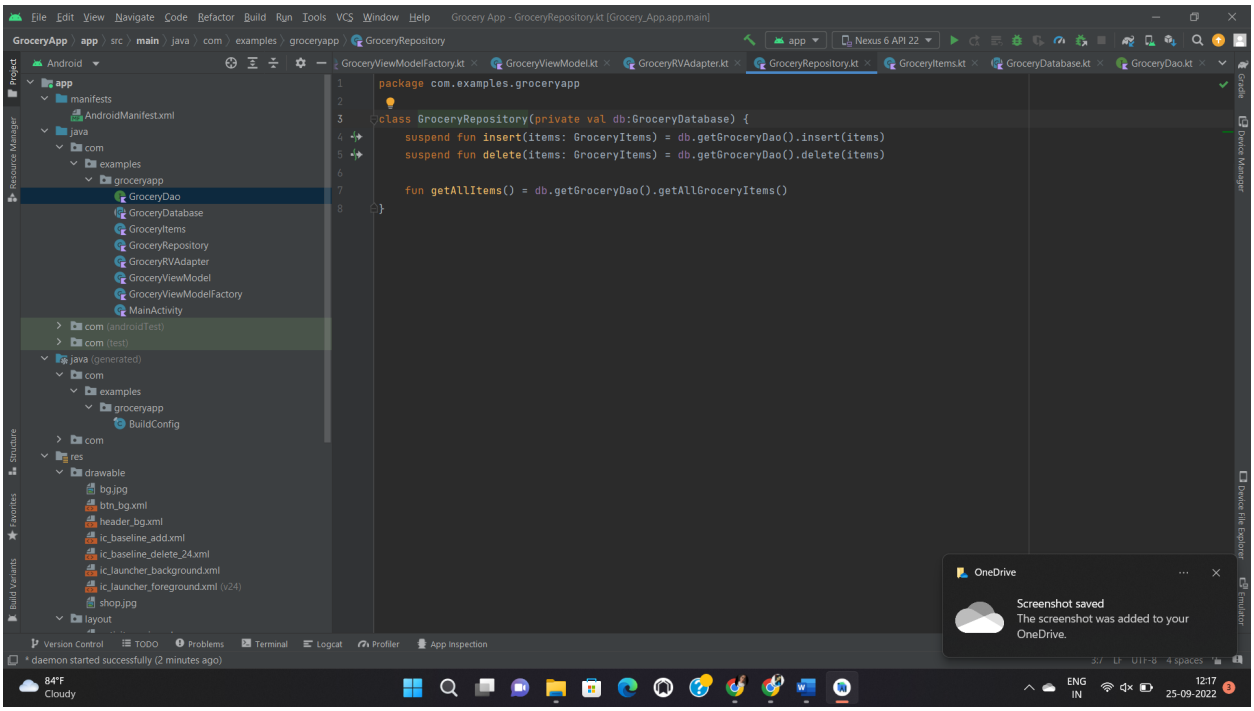


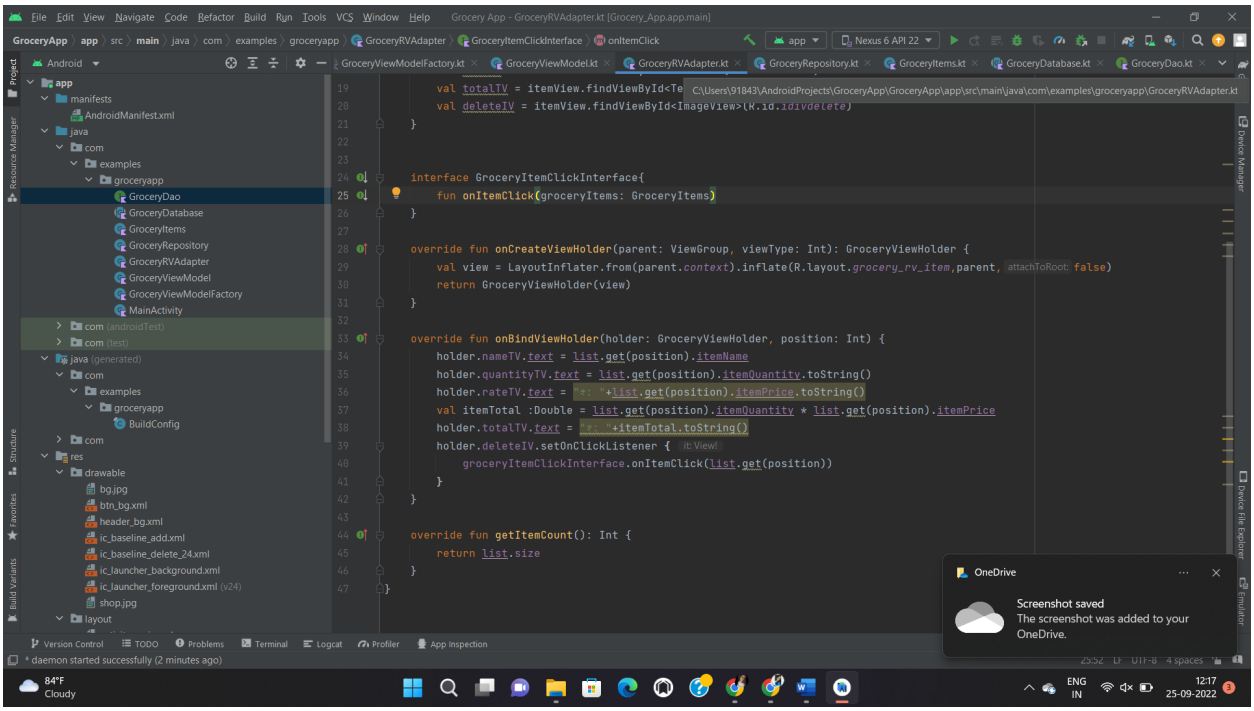
```
package com.examples.groceryapp

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>>


}
```

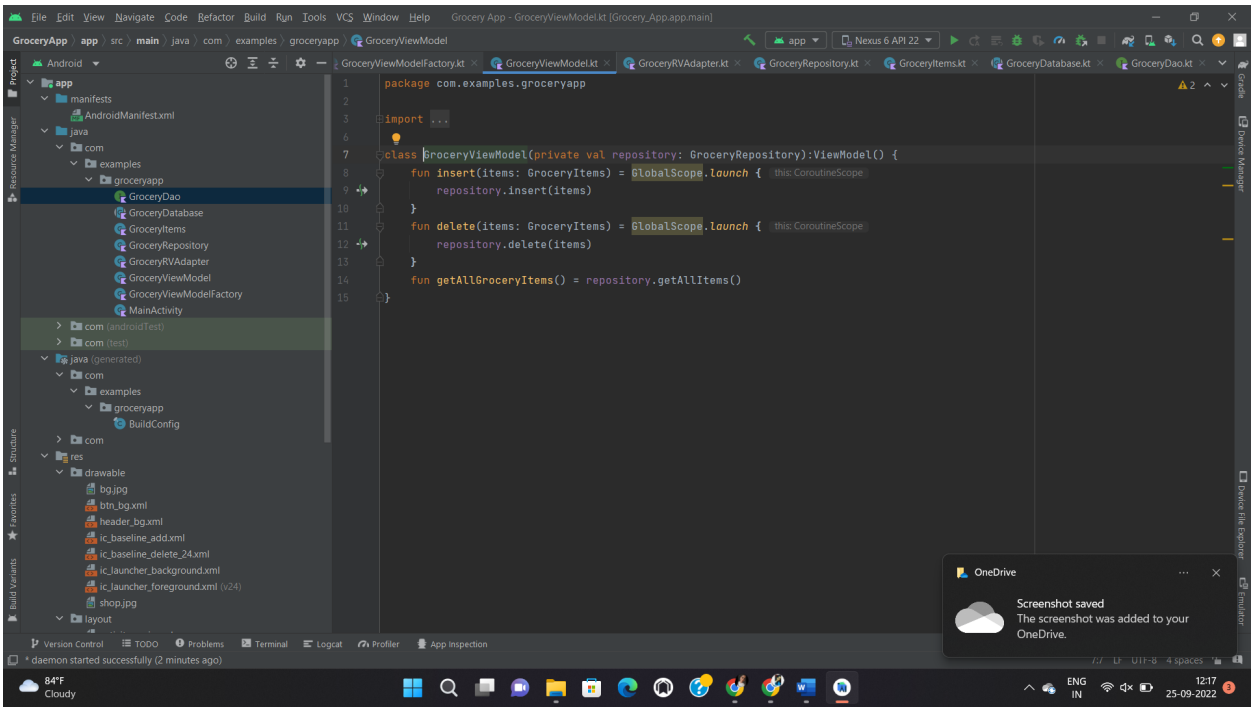


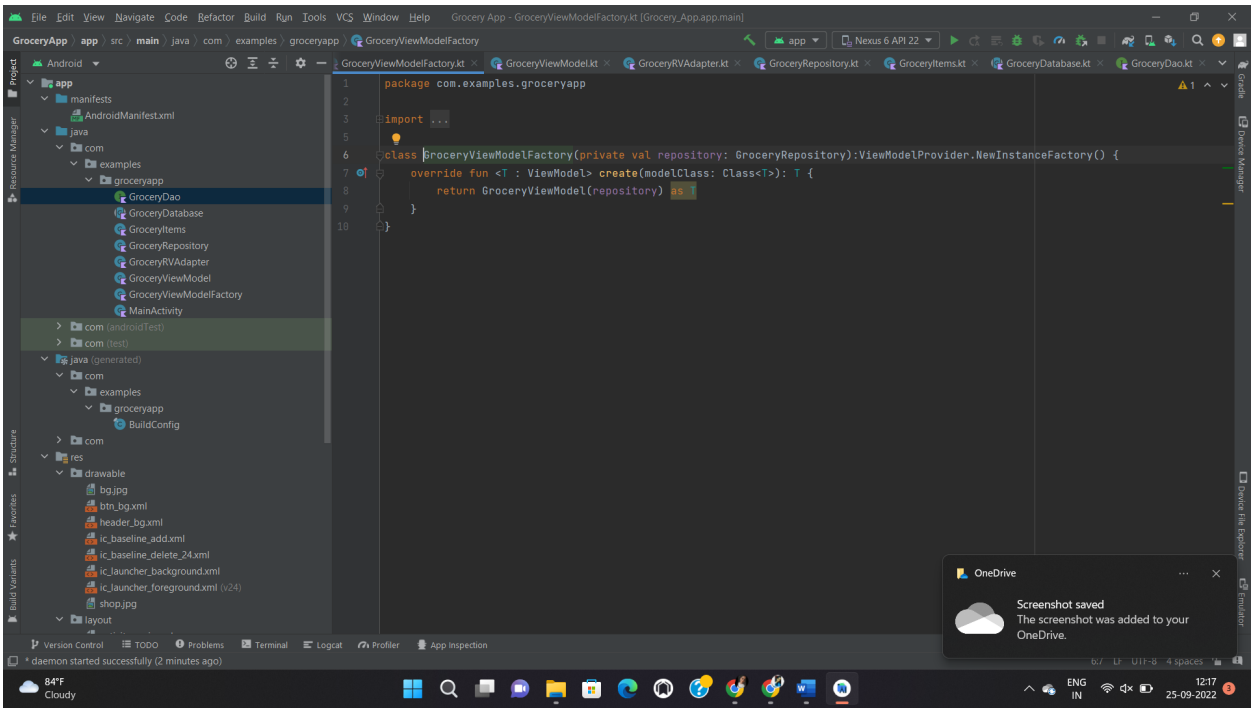
```
package com.examples.groceryapp

import ...


@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,
                "GroceryApp.db"
            ).build()
    }
}
```

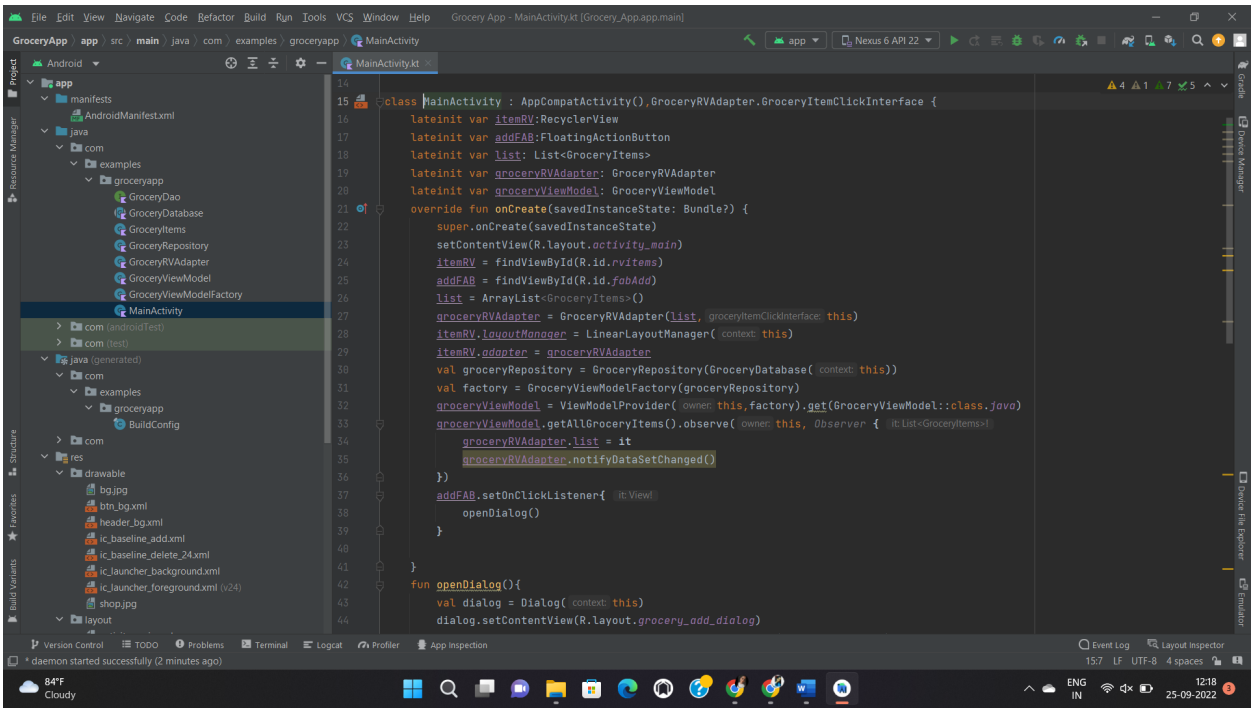

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - GroceryItems.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp GroceryItems
app
Nexus 6 API 22
Android
GroceryViewModelFactory.kt
GroceryViewModel.kt
GroceryRVAdapter.kt
GroceryRepository.kt
GroceryItems.kt
GroceryDatabase.kt
GroceryDao.kt
Run 'app' Shift+F10
app
manifests
AndroidManifest.xml
java
com
examples
groceryapp
GroceryDao
GroceryDatabase
GroceryItems
GroceryRepository
GroceryRVAdapter
GroceryViewModel
GroceryViewModelFactory
MainActivity
com (androidTest)
com (test)
java (generated)
com
examples
groceryapp
BuildConfig
com
res
drawable
bg.jpg
btn_bg.xml
header_bg.xml
ic_baseline_add.xml
ic_baseline_delete_24.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
shop.jpg
layout
package com.examples.groceryapp
import ...
@Entity(tableName = "Grocery_items")
data class GroceryItems (
@ColumnInfo(name = "itemName")
var itemName:String,
@ColumnInfo(name = "itemQuantity")
var itemQuantity:Double,
@ColumnInfo(name = "itemPrice")
var itemPrice:Double,
)
{
@PrimaryKey(autoGenerate = true)
var id:Int?=null
}
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Version Control
TODO
Problems
Terminal
Logcat
Profiler
App Inspection
Run selected configuration
8:12 LF UTF-8 4 spaces
84°F
Cloudy
ENG
IN
12:17
25-09-2022

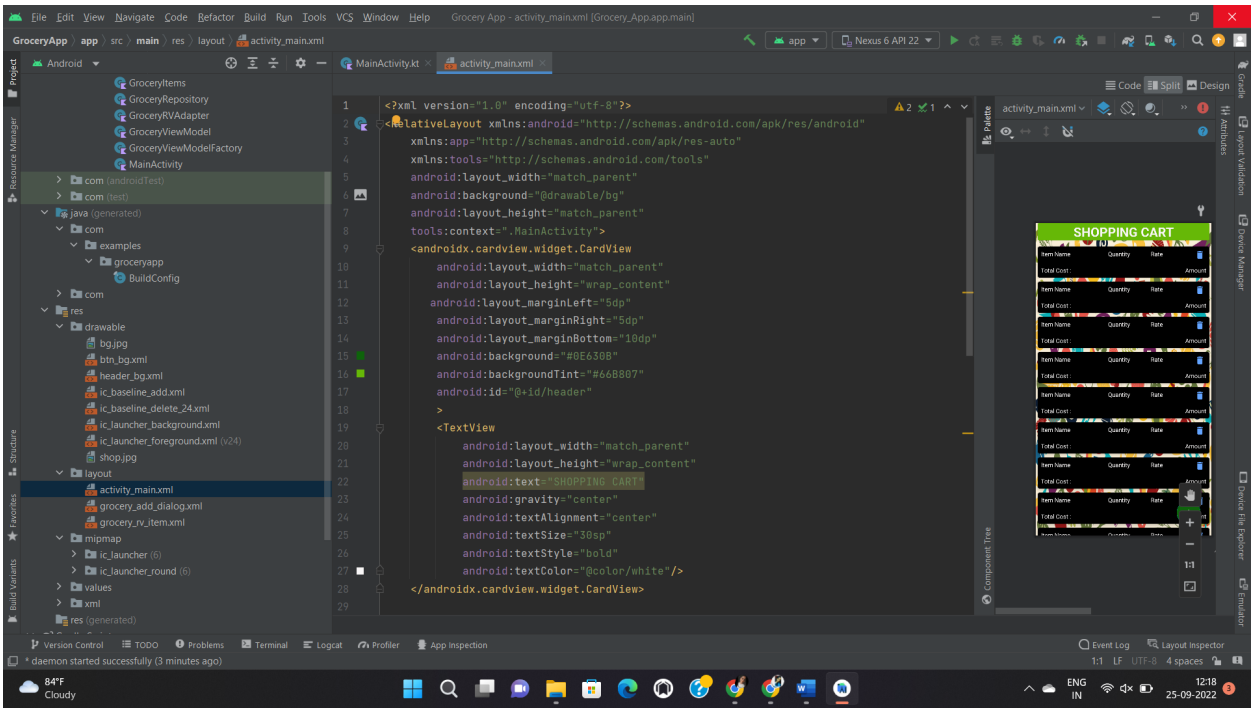

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - GroceryRepository.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp GroceryRepository
app
Nexus 6 API 22
Android
GroceryViewModelFactory.kt
GroceryViewModel.kt
GroceryRVAdapter.kt
GroceryRepository.kt
GroceryItems.kt
GroceryDatabase.kt
GroceryDao.kt
app
manifests
AndroidManifest.xml
java
com
examples
groceryapp
GroceryDao
GroceryDatabase
GroceryItems
GroceryRepository
GroceryRVAdapter
GroceryViewModel
GroceryViewModelFactory
MainActivity
com (androidTest)
com (test)
java (generated)
com
examples
groceryapp
BuildConfig
com
res
drawable
bg.jpg
btn_bg.xml
header_bg.xml
ic_baseline_add.xml
ic_baseline_delete_24.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
shop.jpg
layout
package com.examples.groceryapp
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()
}
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Version Control
TODO
Problems
Terminal
Logcat
Profiler
App Inspection
* daemon started successfully (2 minutes ago)
3:7 LF UTF-8 4 spaces
84°F
Cloudy
ENG
IN
12:17
25-09-2022

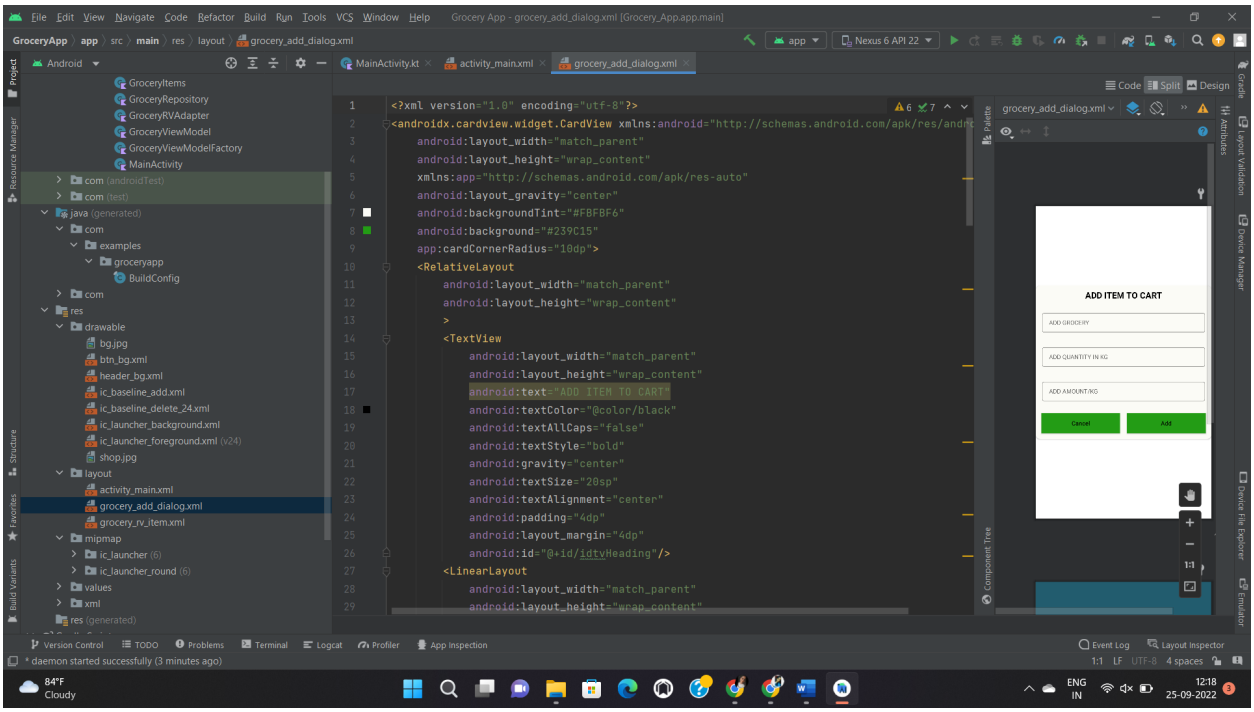

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - GroceryRVAdapter.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp GroceryRVAdapter GroceryItemClickInterface onItemClick
app Nexus 6 API 22
GroceryViewModelFactory.kt GroceryViewModel.kt GroceryRVAdapter.kt GroceryRepository.kt GroceryItems.kt GroceryDatabase.kt GroceryDao.kt
C:\Users\91843\AndroidProjects\GroceryApp\GroceryApp\app\src\main\java\com\examples\groceryapp\GroceryRVAdapter.kt
val totalTV = itemView.findViewById<Te
val deleteIV = itemView.findViewById<ImageView>(R.id.idIvDelete)
}
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, attachToRoot: 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 = "₹: "+list.get(position).itemPrice.toString()
val itemTotal :Double = list.get(position).itemQuantity * list.get(position).itemPrice
holder.totalTV.text = "₹: "+itemTotal.toString()
holder.deleteIV.setOnClickListener { it: View!
groceryItemClickInterface.onItemClick(list.get(position))
}
}
override fun getItemCount(): Int {
return list.size
}
}
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Version Control TODO Problems Terminal Logcat Profiler App Inspection
* daemon started successfully (2 minutes ago)
84°F Cloudy
ENG IN
12:17 25-09-2022

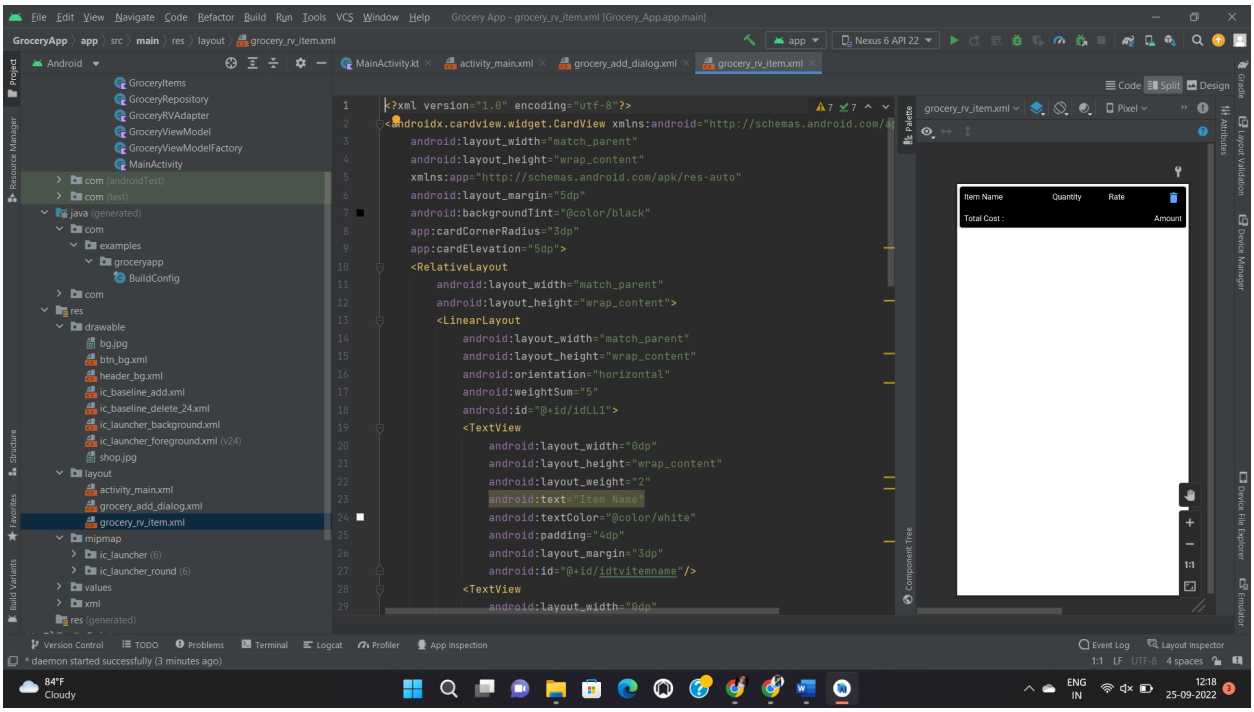

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - GroceryViewModel.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp GroceryViewModel
app Nexus 6 API 22
GroceryViewModelFactory.kt GroceryViewModel.kt GroceryRVAdapter.kt GroceryRepository.kt GroceryItems.kt GroceryDatabase.kt GroceryDao.kt
package com.examples.groceryapp
import ...
class GroceryViewModel(private val repository: GroceryRepository):ViewModel() {
fun insert(items: GroceryItems) = GlobalScope.launch { this: CoroutineScope
repository.insert(items)
}
fun delete(items: GroceryItems) = GlobalScope.launch { this: CoroutineScope
repository.delete(items)
}
fun getAllGroceryItems() = repository.getAllItems()
}
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Version Control TODO Problems Terminal Logcat Profiler App Inspection
* daemon started successfully (2 minutes ago)
84°F Cloudy
ENG IN
12:17 25-09-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - GroceryViewModelFactory.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp GroceryViewModelFactory
Nexus 6 API 22
GroceryViewModelFactory.kt GroceryViewModel.kt GroceryRVAdapter.kt GroceryRepository.kt GroceryItems.kt GroceryDatabase.kt GroceryDao.kt
package com.examples.groceryapp
import ...
class GroceryViewModelFactory(private val repository: GroceryRepository):ViewModelProvider.NewInstanceFactory() {
override fun <T : ViewModel> create(modelClass: Class<T>): T {
return GroceryViewModel(repository) as T
}
}
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Version Control TODO Problems Terminal Logcat Profiler App Inspection
* daemon started successfully (2 minutes ago)
84°F Cloudy
ENG IN
12:17
25-09-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - MainActivity.kt [Grocery_App.app.main]
GroceryApp app src main java com examples groceryapp MainActivity
Nexus 6 API 22
MainActivity.kt
class MainActivity : AppCompatActivity(),GroceryRVAdapter.GroceryItemClickInterface {
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(savedInstanceState)
setContentView(R.layout.activity_main)
itemRV = findViewById(R.id.rvitems)
addFAB = findViewById(R.id.fabAdd)
list = ArrayList<GroceryItems>()
groceryRVAdapter = GroceryRVAdapter(list, groceryItemClickInterface: this)
itemRV.layoutManager = LinearLayoutManager(context: this)
itemRV.adapter = groceryRVAdapter
val groceryRepository = GroceryRepository(GroceryDatabase(context: this))
val factory = GroceryViewModelFactory(groceryRepository)
groceryViewModel = ViewModelProvider(owner: this,factory).get(GroceryViewModel::class.java)
groceryViewModel.getAllGroceryItems().observe(owner: this, Observer { it: List<GroceryItems>
groceryRVAdapter.list = it
groceryRVAdapter.notifyDataSetChanged()
})
addFAB.setOnClickListener{ it: View!
openDialog()
}
}
fun openDialog(){
val dialog = Dialog(context: this)
dialog.setContentView(R.layout.grocery_add_dialog)
Version Control TODO Problems Terminal Logcat Profiler App Inspection
Event Log Layout Inspector
* daemon started successfully (2 minutes ago)
84°F Cloudy
ENG IN
12:18
25-09-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - activity_main.xml [Grocery_App.app.main]
GroceryApp app src main res layout activity_main.xml
app
Nexus 6 API 22
Android
MainActivity.kt
activity_main.xml
Code Split Design
<?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:background="@drawable/bg"
android:layout_height="match_parent"
tools:context=".MainActivity">
<androidx.cardview.widget.CardView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="5dp"
android:layout_marginRight="5dp"
android:layout_marginBottom="10dp"
android:background="#0E630B"
android:backgroundTint="#66B807"
android:id="@+id/header"
>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="SHOPPING CART"
android:gravity="center"
android:textAlignment="center"
android:textSize="30sp"
android:textStyle="bold"
android:textColor="@color/white"/>
</androidx.cardview.widget.CardView>
SHOPPING CART
Version Control TODO Problems Terminal Logcat Profiler App Inspection
Event Log Layout Inspector
daemon started successfully (3 minutes ago)
84°F Cloudy
ENG IN
12:18 25-09-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - grocery_add_dialog.xml [Grocery_App.app.main]
GroceryApp app src main res layout grocery_add_dialog.xml
app
Nexus 6 API 22
Android
MainActivity.kt
activity_main.xml
grocery_add_dialog.xml
Code Split Design
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/andr
android:layout_width="match_parent"
android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_gravity="center"
android:backgroundTint="#FBFBF6"
android:background="#239C15"
app:cardCornerRadius="10dp">
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="ADD ITEM TO CART"
android:textColor="@color/black"
android:textAllCaps="false"
android:textStyle="bold"
android:gravity="center"
android:textSize="20sp"
android:textAlignment="center"
android:padding="4dp"
android:layout_margin="4dp"
android:id="@+id/idtvHeading"/>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
ADD ITEM TO CART
ADD GROCERY
ADD QUANTITY IN KG
ADD AMOUNT/KG
Cancel
Add
Version Control TODO Problems Terminal Logcat Profiler App Inspection
Event Log Layout Inspector
daemon started successfully (3 minutes ago)
84°F Cloudy
ENG IN
12:18 25-09-2022


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
Grocery App - grocery_rv_item.xml [Grocery_App.app.main]
GroceryApp app src main res layout grocery_rv_item.xml
Nexus 6 API 22
Android
GroceryItems
GroceryRepository
GroceryRVAdapter
GroceryViewModel
GroceryViewModelFactory
MainActivity
com (androidTest)
com (test)
java (generated)
com
examples
groceryapp
BuildConfig
com
res
drawable
bg.jpg
btn_bg.xml
header_bg.xml
ic_baseline_add.xml
ic_baseline_delete_24.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
shop.jpg
layout
activity_main.xml
grocery_add_dialog.xml
grocery_rv_item.xml
mipmap
ic_launcher (6)
ic_launcher_round (6)
values
xml
res (generated)
MainActivity.kt
activity_main.xml
grocery_add_dialog.xml
grocery_rv_item.xml
Code Split Design
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/a
android:layout_width="match_parent"
android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_margin="5dp"
android:backgroundTint="@color/black"
app:cardCornerRadius="3dp"
app:cardElevation="5dp">
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="5"
android:id="@+id/idLL1">
<TextView
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="2"
android:text="Item Name"
android:textColor="@color/white"
android:padding="4dp"
android:layout_margin="3dp"
android:id="@+id/idtvitemname"/>
<TextView
android:layout_width="0dp"
grocery_rv_item.xml
Pixel
Item Name
Quantity
Rate
Total Cost :
Amount
Palette
Component Tree
Attributes
Layout Validation
Device Manager
Device File Explorer
Emulator
Version Control
TODO
Problems
Terminal
Logcat
Profiler
App inspection
Event Log
Layout Inspector
* daemon started successfully (3 minutes ago)
1:1 LF UTF-8 4 spaces
84°F
Cloudy
ENG
IN
12:18
25-09-2022

URL's:

Google Developer's Profile:

https://developers.google.com/profile/u/113534802479690885320?authuser=1

GitHub Profile Link:

https://github.com/smartinternz02/SPSGP-104620-Virtual-Internship---Android-Application-Development-Using-Kotlin

Demo Link of project :

https://www.youtube.com/embed/6rMO0mjrzOE

SmartBridge ID or SBID:

SB20220250058

SPS_APL_20220114093

Registered Email:

pratheeksha.19cs068@sode-edu.in

Another Email id :

pratheeksha6666@gmail.com

Acknowledgements

I have taken much efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincerethanksto all of them.

I am highly indebted to SMARTINTERNZ (Experiential Learning & Remote ExternshipPlatform to bring academia & industry very close for a common goal of talent creation) for their guidance and constant supervision as well as for providingnecessary information regarding the

project & also for their support in completing the project. I would like to express my gratitude towards member of (Smart Internz) for their kind co-operation and encouragement which help me in completion of this project.I would like to express my special gratitude and thanks to industry persons for giving me suchattention and time. I would like to convey my heartfelt gratitude to Mr Sandeep Doodigani for his tremendous direction and assistance in the completion of my project.

My thanks and appreciations also go to people who have willingly helped me out with theirabilities.