A Technical Project Report

On

GROCERY APP -USING KOTLIN

IN ANDROID STUDIO

SUBMITTED BY

VEERA LAKSHMI VAIDADI

Email I'd: lakshmivaidadi88@gmail.com

UNDER

SMARTINTERNZ

DATE OF SUBMISSION: 25-SEPT-2022

INDEX

CHAPTER-I

1.1 Abstract:

ANDROID APP KOTLIN has been gaining popularity since it made its debut Android Studio is an open and free which is used to develop the android application for mobile's,TV's,Watches and other Android OS, It provides leverage the power of SQL Database and build a app that shows items added by user.

1.2 Objective:

The goal of the my project is making an app which store the user items in cart and user can modify and delete the added item in list.To develop a reliable system , I have some specific goals such as:

1. ● Develop a system such that user can add item details like product name,product Quantity and Product Price
2. ● Develop a database room which is used to store the user data which already added by the user in cart and user can also remove the previous added item in cart
3. ● Develop the good UI design which user friendly to user
4. ● Develop the good UI which is supported for all android devices

CHAPTER 2

Background & Diagrams

2.1 Background:

Grocery app is a project that will help the user or admin to store the list of items in sequence order.if user want to add extra more items also possible and if user want to remove the previous items also available.

1. ○ UI DESIGN IN ANDROID PLATFORM
2. ○ ANDROID APPLICATION DEVELOPMENT
3. ○ DATABASE CONNECTION TO STORE USER DATA

2.2 Studies:

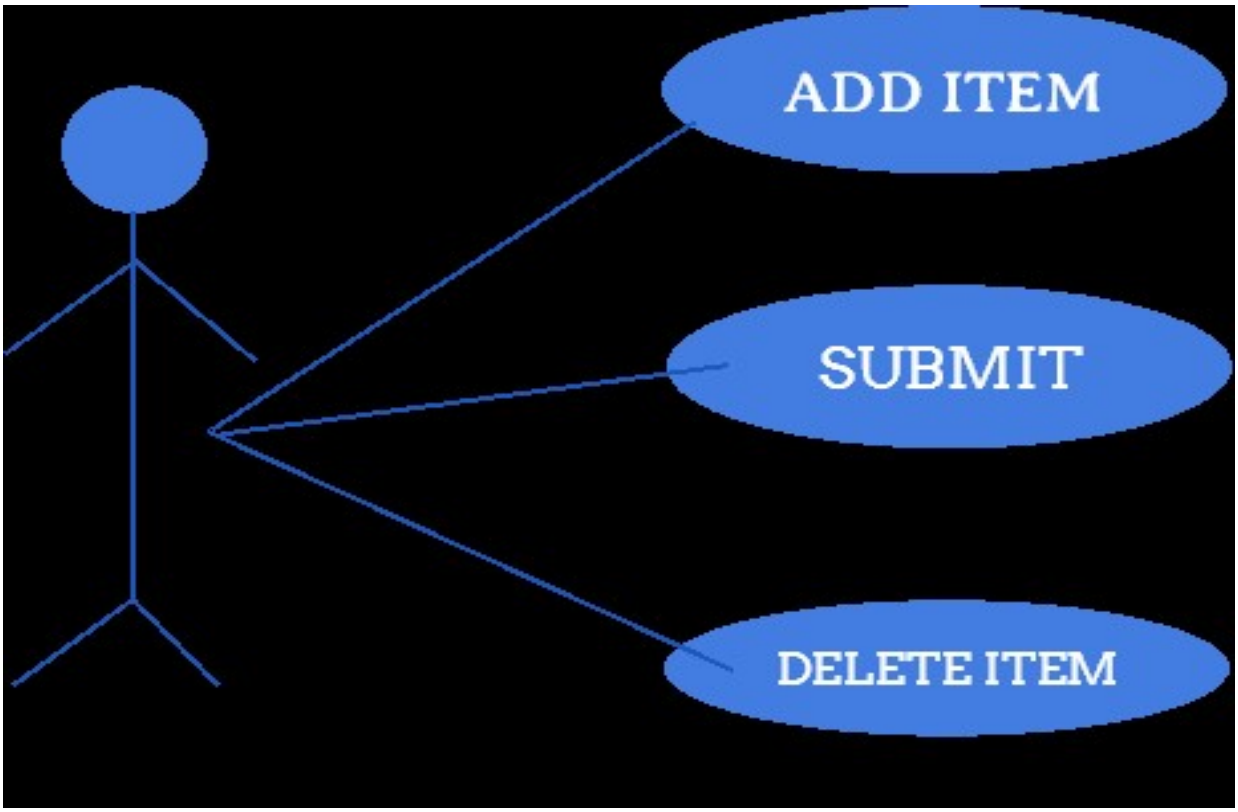
5. ● Design user friendly environment
6. ● Connecting database
7. ● Add material icon for good UI
8. ● Create the vectors and Material Colors

9. ● Navigation of user action from one page to another page

2.3 Challenges:

working with database friendly UI design for UXGradle and SDK settings

2.4 Context Diagram:



CHAPTER 3

REQUIREMENT OF COLLECTION ADN ANALYSIS:

3.1 SOFTWARE:

The Software Package is Developed Using Kotlin and Android Studio.basic Sql Commands Used to Store the Database

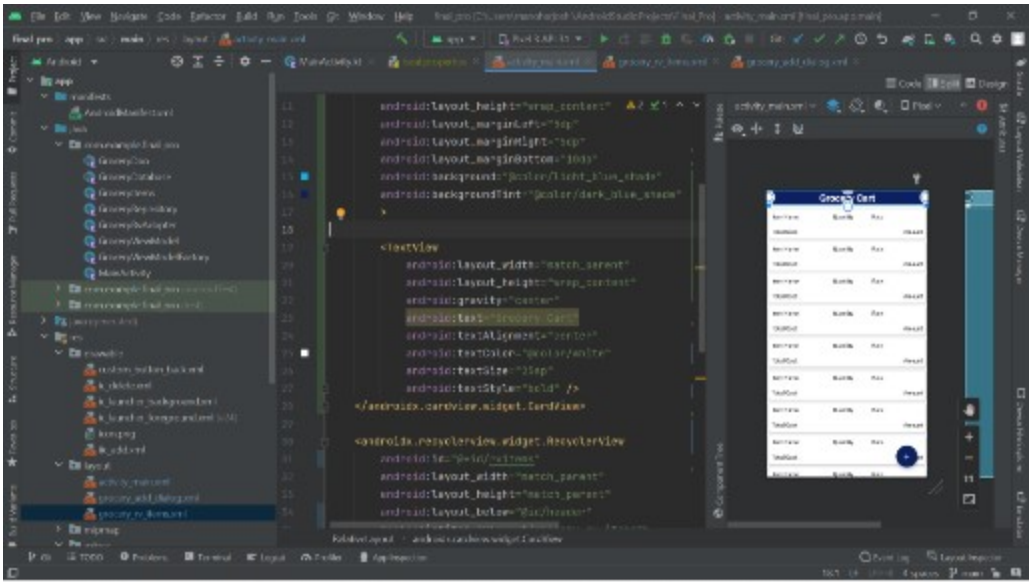
Operating System: Windows 11
Software: Kotlin and Java
Emulator: Pixel 4 (Api 30)

3.2 HARDWARE:

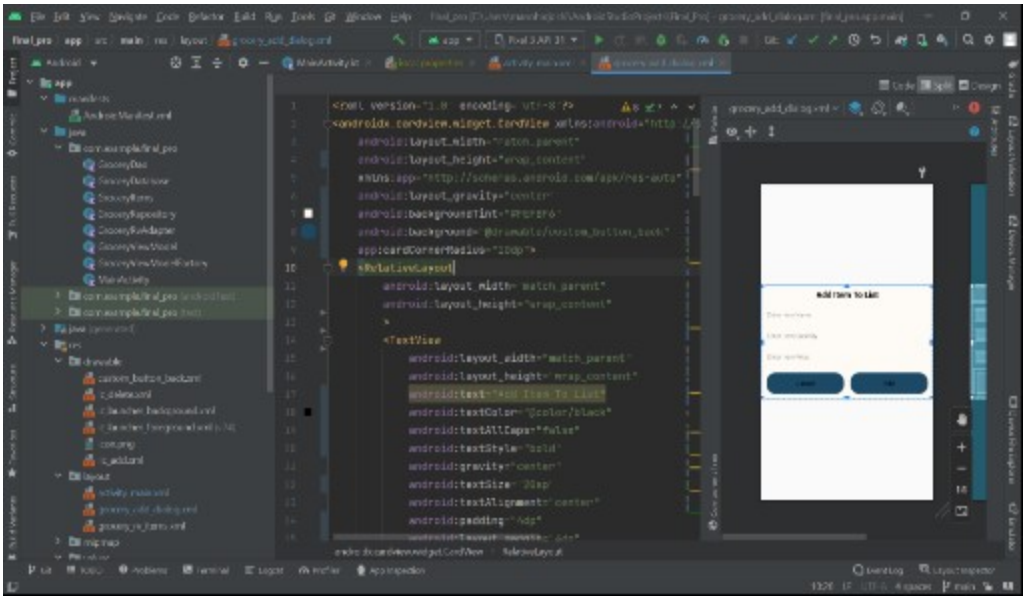
RAM : 8GB RAM
ROM : 20GB

CHAPTER 4

4.1 HOME PAGE & LIST ITEMS:



4.2 POP UP PAGE:



Grocery App Kotlin files

GroceryDao.kt

```
1 package com.rahulpa.groceryapp
2
3 import androidx.lifecycle.LiveData
4 import androidx.room.*
5
6 @Dao
7 interface GroceryDao {
8
9     @Insert(onConflict =
  OnConflictStrategy.REPLACE)
10     fun insert(item: GroceryItems)
11
12     @Delete
13     fun delete(item: GroceryItems)
14
15     @Query("SELECT * FROM grocery_items")
16     fun getAllGroceryItems():
  LiveData<List<GroceryItems>>
17 }
```

GroceryDatabase.kt

```kotlin
package com.rahulpa.groceryapp

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 {
```

```
20                  instance = it
21              }
22          }
23
24          private fun createDatabase(context:
  Context) =
25
  Room.databaseBuilder(context.applicationConte
  xt, GroceryDatabase::class.java,
  "Grocery.db").build()
26      }
27 }
```

GroceryItems.kt

```
1 package com.rahulpa.groceryapp
2
3 import androidx.room.ColumnInfo
4 import androidx.room.Entity
5 import androidx.room.PrimaryKey
6
7 @Entity(tableName = "grocery_items")
8
9 data class GroceryItems(
10
```

```
11      @ColumnInfo(name = "itemName")
12      var itemName: String,
13
14      @ColumnInfo(name = "itemQuantity")
15      var itemQuantity: Int,
16
17      @ColumnInfo(name = "itemPrice")
18      var itemPrice: Int,
19  ) {
20      @PrimaryKey(autoGenerate = true)
21      var id: Int? = null
22  }
```

GroceryRVAdapter.kt

```
1  package com.rahulpa.groceryapp
2
3  import android.view.LayoutInflater
4  import android.view.View
5  import android.view.ViewGroup
6  import android.widget.ImageView
7  import android.widget.TextView
8  import
   androidx.recyclerview.widget.RecyclerView
9
```

```
10 class GroceryRVAdapter(
11     var list: List<GroceryItems>,
12     var groceryItemClickInterface:
  GroceryItemClickInterface
13 ):
  RecyclerView.Adapter<GroceryRVAdapter.Grocery
  ViewHolder>() {
14
15     inner class GroceryViewHolder(itemView:
  View) : RecyclerView.ViewHolder(itemView){
16         val nameTV =
  itemView.findViewById<TextView>(R.id.idTVItem
  Name)
17         val quantityTV =
  itemView.findViewById<TextView>(R.id.idTVQuan
  tity)
18         val rateTV =
  itemView.findViewById<TextView>(R.id.idTVRat
  e)
19         val amountTV =
  itemView.findViewById<TextView>(R.id.idTVTota
  lAmount)
20         val deleteTV =
  itemView.findViewById<ImageView>(R.id.idTVDel
  ete)
21
22     }
```

```
23
24      interface GroceryItemClickInterface{
25          fun onItemClick(groceryItems:
  GroceryItems)
26      }
27
28      override fun onCreateViewHolder(parent:
  ViewGroup, viewType: Int): GroceryViewHolder
  {
29          val view =
  LayoutInflater.from(parent.context).inflate(R
  .layout.grocery_rv_item, parent, false)
30          return GroceryViewHolder(view)
31      }
32
33
34      override fun getItemCount(): Int {
35          return list.size
36      }
37
38      override fun onBindViewHolder(holder:
  GroceryViewHolder, position: Int) {
39          holder.nameTV.text=
  list[position].itemName
40          holder.rateTV.text =
  list[position].itemPrice.toString()
41          holder.quantityTV.text = "Rs.
```

```
  "+list[position].itemQuantity.toString()
42          var itemTotal:Int =
  list[position].itemPrice*list[position].itemQ
  uantity
43
  holder.amountTV.text="Rs."+itemTotal.toString
  ()
44          holder.deleteTV.setOnClickListener{
45
  groceryItemClickInterface.onItemClick(list[po
  sition])
46          }
47
48      }
49
50 }
```

GroceryRepository.kt

```
1 package com.rahulpa.groceryapp
2
3 class GroceryRepository(private val db:
  GroceryDatabase) {
4
5     suspend fun insert(item: GroceryItems) =
  db.getGroceryDao().insert(item)
```

```
6     suspend fun delete(item: GroceryItems) =
  db.getGroceryDao().delete(item)
7
8     fun allGroceryItems() =
  db.getGroceryDao().getAllGroceryItems()
9 }
```

GroceryViewModel.kt

```
1 package com.rahulpa.groceryapp
2
3 import androidx.lifecycle.ViewModel
4 import kotlinx.coroutines.GlobalScope
5 import kotlinx.coroutines.launch
6
7 class GroceryViewModel(private val
  repository: GroceryRepository) : ViewModel()
  {
8
9
10     fun insert(item: GroceryItems) =
  GlobalScope.launch {
11         repository.insert(item)
12     }
13
```

```
14
15      fun delete(item: GroceryItems) =
  GlobalScope.launch {
16          repository.delete(item)
17      }
18
19
20      fun allGroceryItems() =
  repository.allGroceryItems()
21
22 }
```

GroceryViewModelFactory.kt

```
1 package com.rahulpa.groceryapp
2
3 import androidx.lifecycle.ViewModel
4 import androidx.lifecycle.ViewModelProvider
5
6 class GroceryViewModelFactory(private val
  repository: GroceryRepository):
  ViewModelProvider.NewInstanceFactory() {
7
8     override fun <T : ViewModel>
  create(modelClass: Class<T>): T {
9         return GroceryViewModel(repository)
```

```
  as T
10     }
11 }
```

MainActivity.kt

```
1 package com.rahulpa.groceryapp
2
3 import android.app.Dialog
4 import
  androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.widget.Button
7 import android.widget.EditText
8 import android.widget.Toast
9 import androidx.lifecycle.Observer
10 import androidx.lifecycle.ViewModelProvider
11 import
  androidx.recyclerview.widget.LinearLayoutMana
  ger
12 import
  androidx.recyclerview.widget.RecyclerView
13 import
  com.google.android.material.floatingactionbut
  ton.FloatingActionButton
14
```

```
15 class MainActivity :
  AppCompatActivity(),GroceryRVAdapter.GroceryI
  temClickInterface {
16      lateinit var itemsRV: RecyclerView
17      lateinit var addFAB:
  FloatingActionButton
18      lateinit var list: List<GroceryItems>
19      lateinit var groceryRVAdapter:
  GroceryRVAdapter
20      lateinit var
  grocerViewModal:GroceryViewModel
21
22      override fun
  onCreate(savedInstanceState: Bundle?) {
23          super.onCreate(savedInstanceState)
24
  setContentView(R.layout.activity_main)
25          itemsRV=findViewById(R.id.idRVItems)
26          addFAB=findViewById(R.id.idFABAdd)
27          list=ArrayList<GroceryItems>()
28          groceryRVAdapter =
  GroceryRVAdapter(list,this)
29
  itemsRV.layoutManager=LinearLayoutManager(thi
  s)
30          itemsRV.adapter = groceryRVAdapter
31          val groceryRepository =
```

```
  GroceryRepository(GroceryDatabase(this))
32          val factory =
  GroceryViewModelFactory(groceryRepository)
33          grocerViewModal =
  ViewModelProvider(this,factory).get(GroceryVi
  ewModel::class.java)
34
  grocerViewModal.allGroceryItems().observe(thi
  s, Observer {
35              groceryRVAdapter.list=it
36
  groceryRVAdapter.notifyDataSetChanged()
37          })
38          addFAB.setOnClickListener{
39              openDialog()
40          }
41
42      }
43
44      fun openDialog(){
45          var dialog = Dialog(this)
46
  dialog.setContentView(R.layout.grocery_dialo
  g)
47          val cancelBtn =
  dialog.findViewById<Button>(R.id.idCancelButt
  on)
```

```
48          val addBtn =
  dialog.findViewById<Button>(R.id.idAddButton)
49          val itemEdt =
  dialog.findViewById<EditText>(R.id.idEditItem
  Name)
50          val itemPriceEdt =
  dialog.findViewById<EditText>(R.id.idEditItem
  Price)
51          val itemQtyEdt =
  dialog.findViewById<EditText>(R.id.idEditItem
  Qty)
52          cancelBtn.setOnClickListener{
53              dialog.dismiss()
54          }
55          addBtn.setOnClickListener {
56              val itemName:String =
  itemEdt.text.toString()
57              val itemPrice:String =
  itemPriceEdt.text.toString()
58              val itemQty:String =
  itemQtyEdt.text.toString()
59              val qty:Int = itemQty.toInt()
60              val pr : Int = itemPrice.toInt()
61
62 if(itemName.isNotEmpty()&&itemPrice.isNotEmp
  ty()&&itemQty.isNotEmpty()){
63                  val items =
```

```
  GroceryItems(itemName,qty,pr)
64
  grocerViewModal.insert(items)
65
  Toast.makeText(applicationContext,"Item
  Inserted...",Toast.LENGTH_SHORT).show()
66
  groceryRVAdapter.notifyDataSetChanged()
67                    dialog.dismiss()
68              }else{
69
  Toast.makeText(applicationContext,"Enter all
  data",Toast.LENGTH_SHORT).show()
70              }
71
72          }
73          dialog.show()
74      }
75
76      override  fun onItemClick(groceryItems:
  GroceryItems){
77          grocerViewModal.delete(groceryItems)
78
  groceryRVAdapter.notifyDataSetChanged()
79
  Toast.makeText(applicationContext,"Item
  Deleated...",Toast.LENGTH_SHORT).show()
```

```
80      }
81 }
```

CHAPTER 5

5.1 Conclusion:

This application will help to store the list of items date include name price and quantity.most of shopping admins want to store the his data in list, the application is very help full for them.not only in shops and admins also helpful for user what they buy and what they want to buy.

5.2 Future Scope:

This application help to store the list of items by Admin .In Future we can also add user panel which is add by user required item that are submitted to admin

The Feature are:

1. ● Add User Panel
2. ● Add Admin Panel
3. ● Provide Login authentication
4. ● Add image to user product and rating

GIT HUB link: https://github.com/Lakshmiveeru

DEMO link:
https://drive.google.com/file/d/1PmoksfjyIDSBM88LGvMrPe8yUudrXjp_/view?usp=sharing