Name : Vishal N M  Virtual Internship id:  SPS_APL_20220069689

# SMARTBRIDE-SMART INTERZ

Google Supported Virtual Internship - Android Application Development using Kotlin

## INTRODUCTION

Overview: In this project, we are going to build a grocery application in android using android studio.

Purpose: Grocery Application Using Kotlin

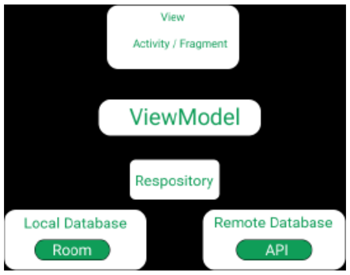
## LITERARY SURVEY

Existing Problem: Many times we forget to purchase things that we want to buy, after all, we can’t remember all the items, so with the help of this app, you can note down your grocery items that you are going to purchase, by doing this you can’t forget any items that you want to purchase.

Proposed Solution: A sample image is given below to get an idea about what we are going to do in this article. Note that we are going to implement this project using the Kotlin language. In this project, we are using MVVM (Model View ViewModel) for architectural patterns, Room for database, Coroutines and RecyclerView to display the list of items.

## THEORETICAL ANALYSIS

Block Diagram:



Hardware Requirements: A computer or a laptop with a good internet connection and a android device.(Preferably running the latest software).

Software Requirements: Android Studio software, Windows or Mac operating System and a browser to download required files.

## EXPERIMENTAL INVESTIGATIONS

### MVVM (Model View ViewModel)

MVVM architecture in android is used to give structure to the project's code and understand code easily. MVVM is an architectural design pattern in android. MVVM treat Activity classes and XML files as View. This design pattern completely separate UI from its logic. Here is an image to quickly understand MVVM.

### ROOM DataBase

Room 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.

### RecycleView

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

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.

## FLOWCHART

Step 1: Create a New Project

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

Step 3: Implement room database

Step 4: Now we will implement the architectural structure in the app

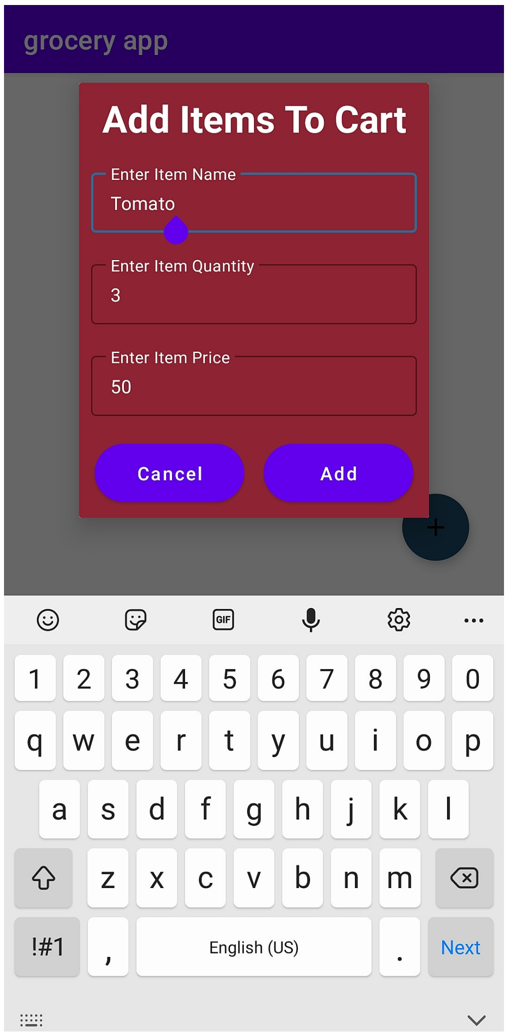
Step 5: Now let's jump into the UI part

Step 6: Let's implement RecyclerView.

Step 7: To enter grocery item, quantity, and price from the user we have to create an interface

Step 8: In this step finally we will code in our MainActivity

RESULTS ALONG WITH SCREENSHOTS

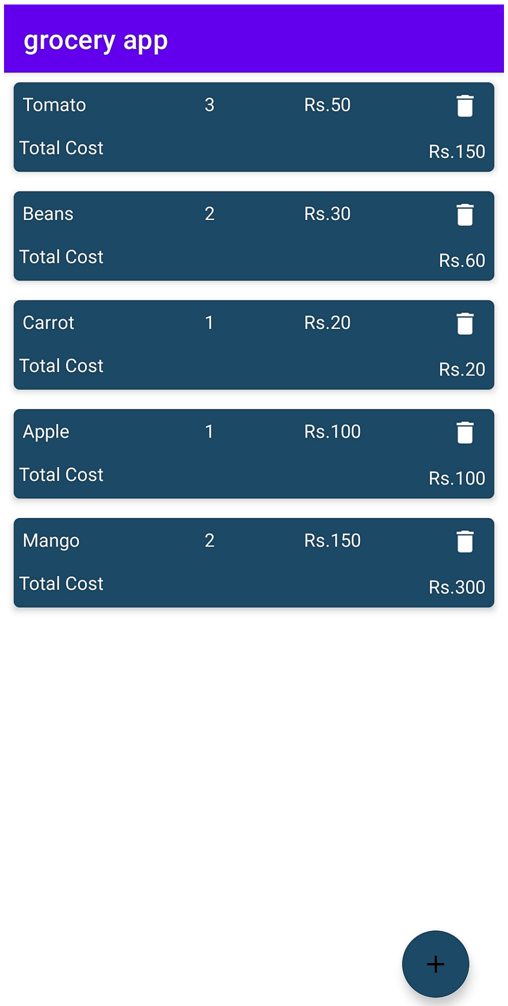

grocery app

Add Items To Cart

Enter Item Name: Tomato

Enter Item Quantity: 3

Enter Item Price: 50


grocery app

| | | | |
|---|---|---|---|
| Tomato | 3 | Rs.50 | |
| Total Cost | | | Rs.150 |
| Beans | 2 | Rs.30 | |
| Total Cost | | | Rs.60 |
| Carrot | 1 | Rs.20 | |
| Total Cost | | | Rs.20 |
| Apple | 1 | Rs.100 | |
| Total Cost | | | Rs.100 |
| Mango | 2 | Rs.150 | |
| Total Cost | | | Rs.300 |

## ADVANTAGES AND DISADVANTAGES

Advantages: We could easily buy grocery or any other item without remembering the amount or price of the item by using this grocery application built using kotlin.

Disadvantages: The app is not very feature rich and we have manually add and delete the items which we require.

## APPLICATIONS

This application can be used in every household to buy grocery items at their convenience without remembering the items.

This app can also used to add other items like stationary items, clothes, etc .

## CONCLUSION

In this project we have built a grocery application in android using android studio. Many times we forget to purchase things that we want to buy, after all, we can’t remember all the items, so with the help of this app, you can note down your grocery items that you are going to purchase, by doing this you can’t forget any items that you want to purchase. We have implemented the project using the Kotlin language.In this project, we are using MVVM (Model View ViewModel) for architectural patterns, Room for database, Coroutines and RecyclerView to display the list of items.

## FUTURE SCOPE

We can extend the functionality of the app by giving a feature of generating a bill at the end. We can also improve the user interface by using images of the items that we add. By connecting with the notification of the android device we can automatically delete the item if the item has been bought by recognizing the amount paid using the upi network. We can also add a feature where in which we can add items using our voice.

## BIBLIOGRAPHY

GeeksforGeeks: https://www.geeksforgeeks.org/

Android Studio: https://developer.android.com/studio?gclid=CjwKCAjw4c-ZBhAEEiwAZ105RQfb47RI62euACSluQRFbFh8VQKy25qz8ALhUgPavJaE2HqANCvcPhoCltUQAvD_BwE&gclsrc=aw.ds

Smartbridge: https://www.thesmartbridge.com/

Android developers: https://developer.android.com/

APPENDIX

Source Code:

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'kotlin-kapt'

android {
   compileSdkVersion 29
   buildToolsVersion "30.0.3"

   defaultConfig {
      applicationId "com.example.grocerylist"
      minSdkVersion 16
      targetSdkVersion 29
      versionCode 1
      versionName "1.0"

      testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
   }

   buildTypes {
      release {
         minifyEnabled false
         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-
rules.pro'
      }
   }
   compileOptions {
      sourceCompatibility = 1.8
      targetCompatibility = 1.8
   }
   kotlinOptions {
      jvmTarget = "1.8"
```

```
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.core:core-ktx:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

    def room_version = "2.2.1"
    def lifecycle_version = "2.0.0"

    // Room and Architectural Components
    implementation "androidx.room:room-runtime:$room_version"
    implementation "androidx.legacy:legacy-support-v4:1.0.0"
    implementation 'androidx.lifecycle:lifecycle-extensions:2.1.0'
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.1.0'
    implementation "androidx.room:room-ktx:2.2.1"
    kapt "androidx.room:room-compiler:$room_version"

    // Coroutines
    implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.0'
    implementation "org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.0"

    // New Material Design
    implementation "com.google.android.material:material:1.0.0"

    // ViewModel
    implementation "androidx.lifecycle:lifecycle-extensions:$lifecycle_version"
    implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:$lifecycle_version"
    kapt "androidx.lifecycle:lifecycle-compiler:$lifecycle_version"

}
```

```xml
<resources>
    <string name="app_name">Fresh Basket</string>

    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>
    <string name="itemName">Banana</string>
    <string name="itemQuantity">35</string>
    <string name="itemPrice">250Rs</string>
    <string name="totalCost">20</string>
    <string name="totalCostTitle">Total Cost</string>
    <string name="title">Add Items to your cart</string>
    <string name="etItem">Item</string>
    <string name="etQuantity">Quantity</string>
    <string name="etPrice">Price</string>
    <string name="save">Save</string>
    <string name="cancel">Cancel</string>

</resources>

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#0AD042</color>
    <color name="colorPrimaryDark">#03551A</color>
    <color name="colorAccent">#03DAC5</color>
    <color name="black">#000000</color>
    <color name="white">#ffffff</color>
</resources>
```

```kotlin
package com.example.grocerylist.Database.Entity

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

```
// This is a data class which store data.
// Entities class create a table in database,
// in our database we will create three column

@Entity(tableName = "grocery_items")

data class GroceryItems(

    // create itemName variable to
    // store grocery items.
    @ColumnInfo(name = "itemName")
    var itemName: String,

    // create itemQuantity variable
    // to store grocery quantity.
    @ColumnInfo(name = "itemQuantity")
    var itemQuantity: Int,

    // create itemPrice variable to
    // store grocery price.
    @ColumnInfo(name = "itemPrice")
    var itemPrice: Int
) {
    // Primary key is a unique key
    // for different database.
    @PrimaryKey(autoGenerate = true)
    var id: Int? = null
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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="#ffffff"
android:orientation="vertical"
tools:context=".UI.MainActivity">

<!-- To create a app bar with logo image. -->
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:src="@drawable/logo"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

<!-- In this image view we will add a title image -->
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="35dp"
    android:src="@drawable/title"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView2"
    app:layout_constraintVertical_bias="0.0" />

<!-- Recycler View to display list -->
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvList"
    android:layout_width="match_parent"
    android:layout_height="470dp"
    android:background="@color/white"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="1.0">

    </androidx.recyclerview.widget.RecyclerView>

    <!-- This button is used to open dialog box in
         which user can enter grocery items -->
    <Button
        android:id="@+id/btnAdd"
        android:layout_width="60dp"
        android:layout_height="wrap_content"
        android:background="@drawable/button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.954"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.969" />

</androidx.constraintlayout.widget.ConstraintLayout>
```