PROJECT REPORT

Grocery List

(Application Development in Kotlin using Android Studio)

Under

By:
Kanaishk Garg
SPS_APL_20220100318

INTRODUCTION

- **Overview**

Whenever we go to shop something one of the things, we commonly do is make a list of items we intend to buy as it is common to forget something if we do not. But an issue with paper lists is that they can be spoiled, torn or lost easily. Also, we need to carry pen to mark what is bought already. This is both cumbersome and a waste of resources.

- **Purpose**

This is a listing app that tries to resolve this problem. You can enter what you want to buy, in what quantity and its price. It also computes the total price that item will cost. This allows us to reduce clutter as we only need to carry out phone with us to have the list with us.

LITERATURE SURVEY

- **Existing problem**

The main aim of this project is to list the items so that whenever users go to grocery stores, users will not be able to forget their items and this grocery application helps the users to tackle their day-to-day chaos more effortlessly. It's not easy for the users to remember every item in this hectic lifestyle, they frequently can't recall their required necessity so we decided to build an app to store the items in the database for their future use. After buying the items users can delete the added items in the database.

- **Proposed solution**

The goal of this project is to make an app that stores the user items in a cart and can modify and delete the added item in the list. To develop a reliable system, I have some specific goals such as:

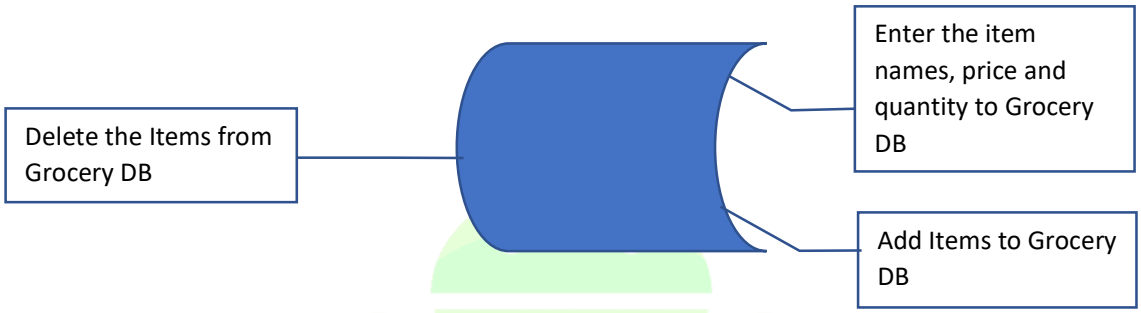
- Develop a system such that users can add item details like product name, product Quantity, and Product Price.
- Develop a database room that is used to store the user data which already been added by the user in the cart and the user can also remove the previously added item in the cart.
- Develop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

THEORITICAL ANALYSIS

- **Block diagram**

The grocery cart application project will help the user or admin to store the list of items in proper sequence. User/Admin can add and remove the items in the list according to his/her will.

- UI DESIGN IN THE ANDROID PLATFORM
- ANDROID APPLICATION DEVELOPMENT
- DATABASE CONNECTION TO STORE USER DATA



- **Hardware / Software designing**

The Software Package is developed using Kotlin and Android Studio, basic SQL commands are used to store the database. Development requirement:

Operating System:	**Windows 10 (x64)**
Software:	**Kotlin**
Emulator:	**Pixel 5 API 31**
RAM:	**16 GB**
ROM:	**20 GB**

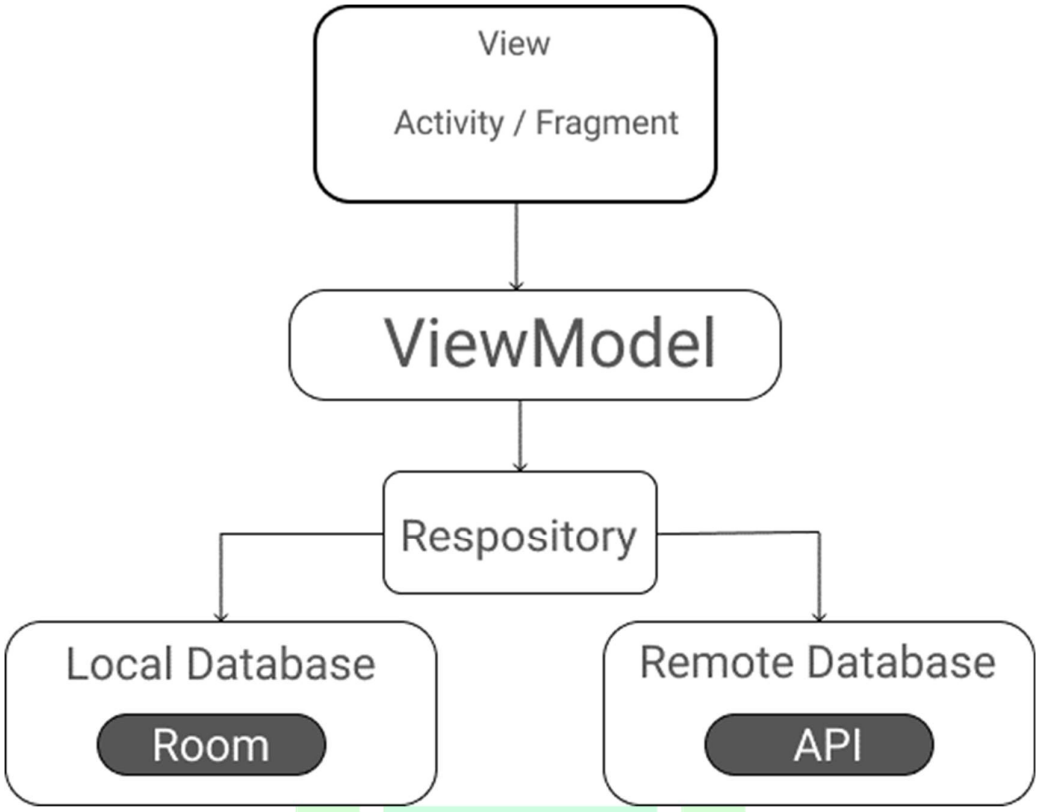
EXPERIMENTAL INVESTIGATIONS

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

- **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.

- **RecyclerView**

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 coroutines to perform an operation on other threads, by this our main thread doesn't block and our app doesn't crash.

Step-by-Step process:

Step 1: Create a New Project

To create a new project in Android Studio please refer to How to Create/Start a New Project in 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: Implement Room Database

a) Entities class

The entities class contains all the columns in 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 PrimaryKey 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 functions that we want to implement on the database. This interface also 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 database and annotated with @Insert, second is for deleting items from the database annotated with @Delete and Third is for getting all items annotated with @Query. Go to the **app > java > com.example.application-name**. Rightclick 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 entities are the entities array list all the data entities associating with the database and version shows the current version of

the database. This database class inherits from the Room Database class. In **GroceryDatabase** class we will make an abstract method to get an instance of DAO and further use this method from the DAO instance to interact with the database. Go to the **app > java > com.example.application-name**. Right-click on **com.example.application-name** go to new and create Kotlin file/class as **GroceryDatabase**.

Step 4: Now we will implement the Architectural Structure in the App

a) Repository class

The repository is one of the design structures. The repository class gives the data to the ViewModel class and then the ViewModel class uses that data for Views. The repository will choose the appropriate data locally or on the network. Here in our Grocery Repository class data fetch locally from the Room database. We will add constructor value by creating an instance of the database and stored in the db variable in the Grocery Repository class. Go to the **app > java > com.example.application-name**. Right-click on **com.example.application-name** go to new and create Kotlin file/class as **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 then right-click on UI package and create a Kotlin file/class.

b) ViewModel class

ViewModel class used as an interface between View and Data. Grocery View Model class inherit from View Model class and we will pass constructor value by creating instance variable of Repository class and stored in repository variable. 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 and create a Kotlin file/class and name the file as **GroceryViewModel**.

c) FactoryViewModel class

We will inherit the Grocery ViewModel Factory 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, item quantity, 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 recycler view. In the GroceryAdapter class, we will add constructor value by storing entities class as a list in list variable and create an instance of the view model. In Grocery Adapter we will override three functions: onCreateViewHolder, getItemCount, and onbindViewHolder, we will also create an inner class called grocery view holder. Go to the **app > java > com.example.applicationname**. 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 name it **GroceryAdapter**.

Step 7:

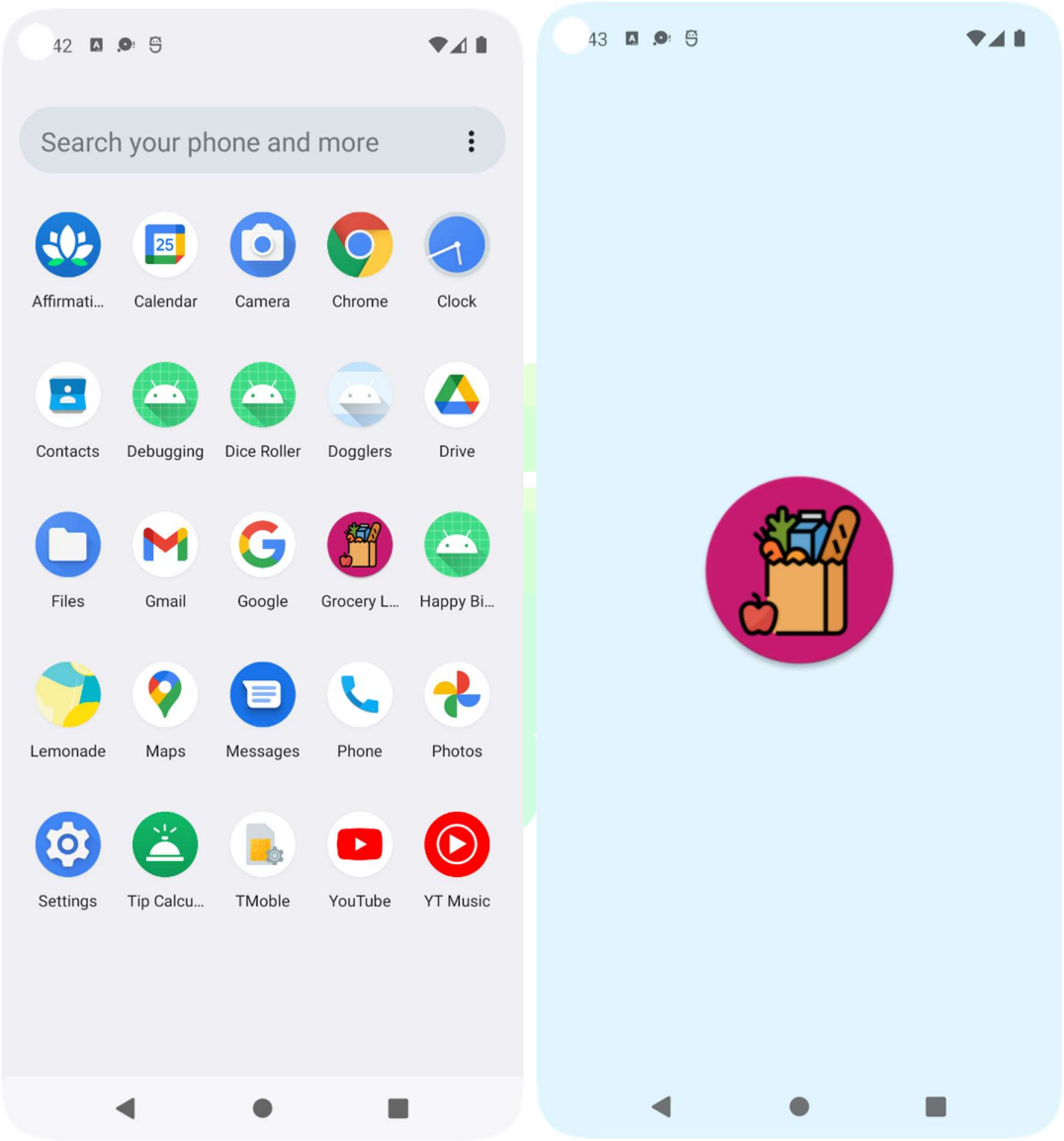
To enter grocery item, quantity, and price from the user we have to create an interface. To implement this interface, we will use DialogBox. First create UI of dialog box. In this dialog box we will add three edit text and two text view. Three edit text to enter grocery item name, quantity and price. Two text view one for save and other for cancel. After clicking the save text all data saved into the database and 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.applicationname > UI.** Right-click on the **UI** package and create a Kotlin file/class and create an **interface** name it as **DialogListener**.

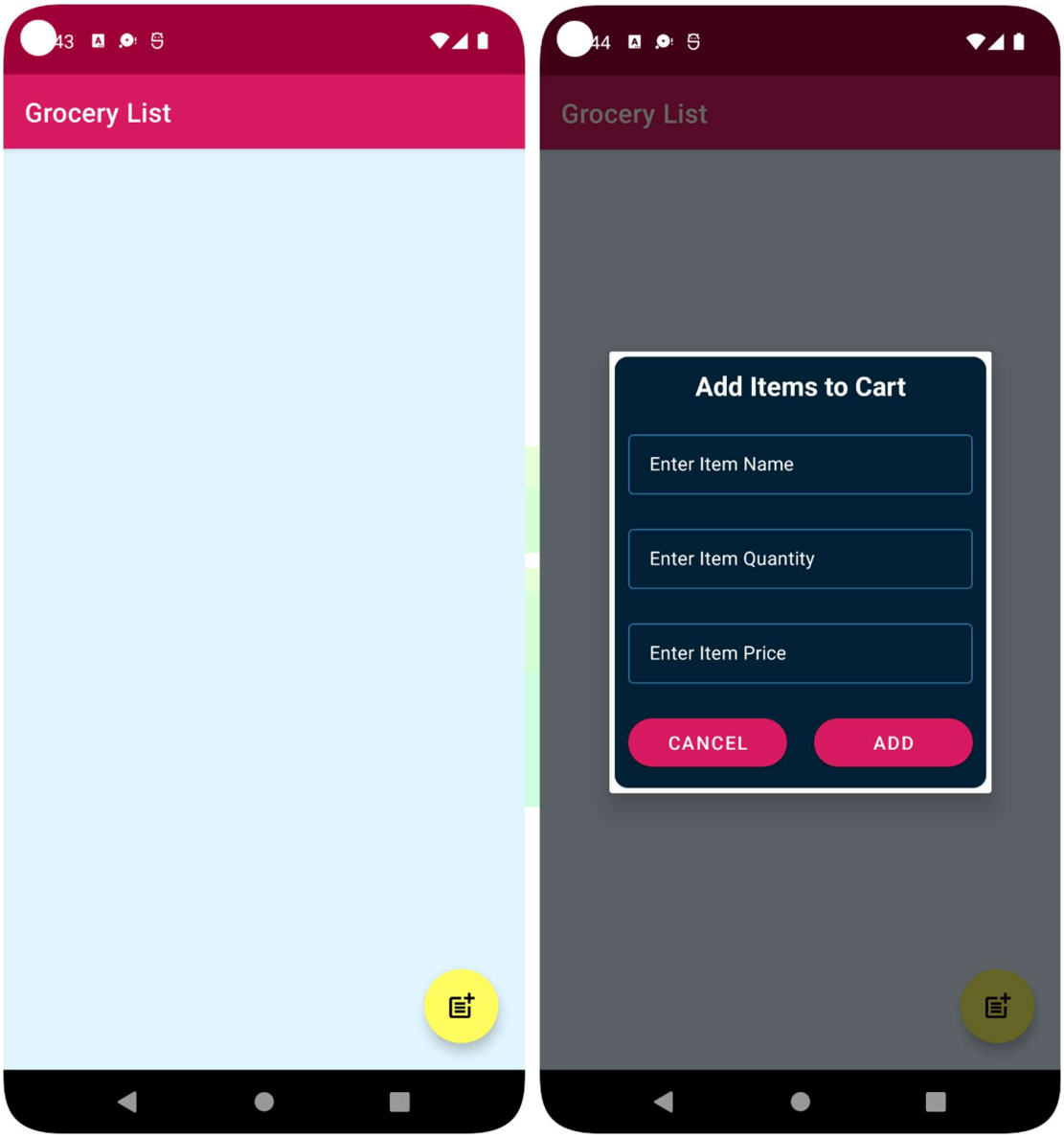
Step 8:

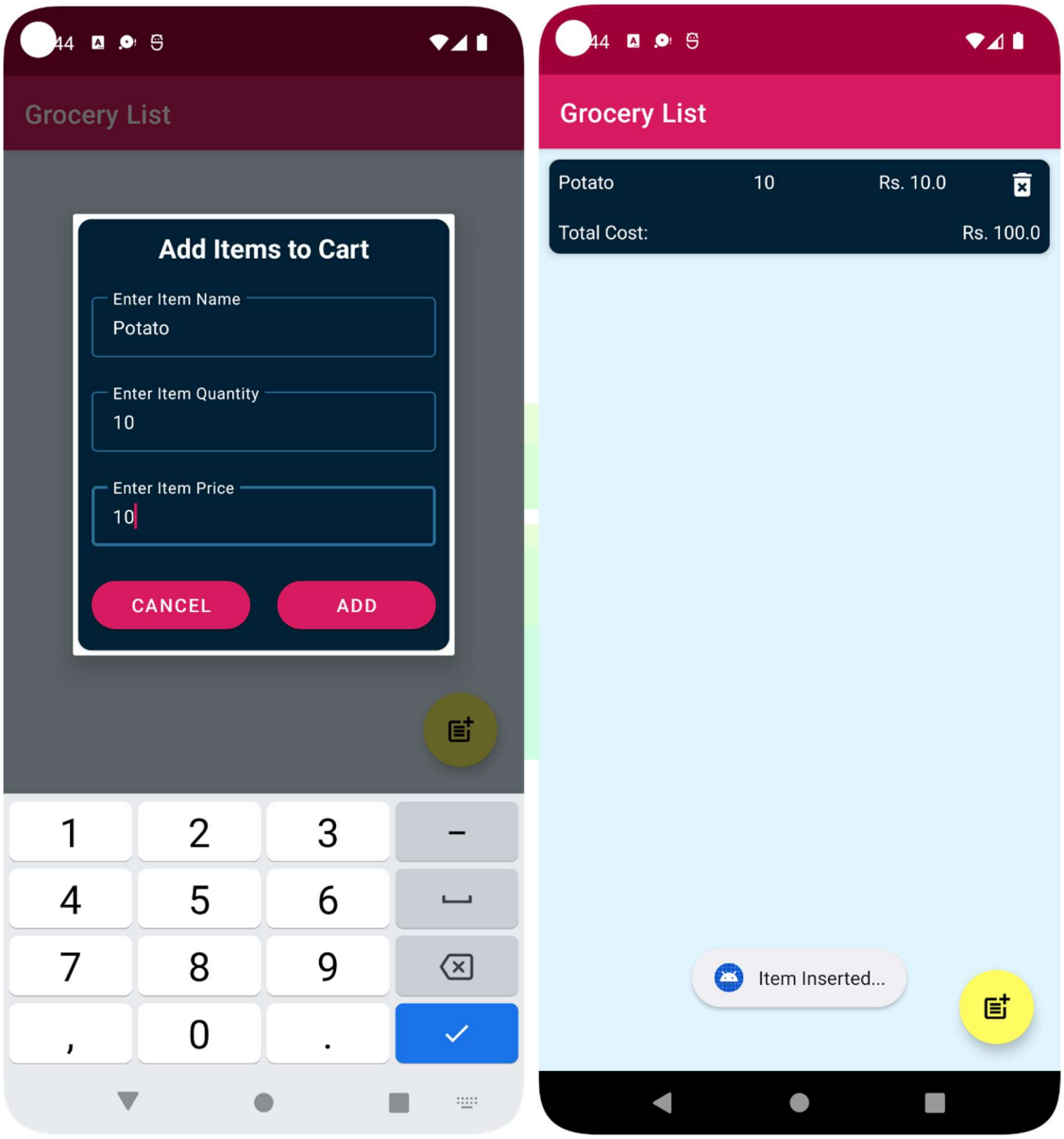
In this final step we will code in our **MainActivity**. In our **MainActivity**, we have to set up the recycler view and add click listener on add button to open the dialog box.

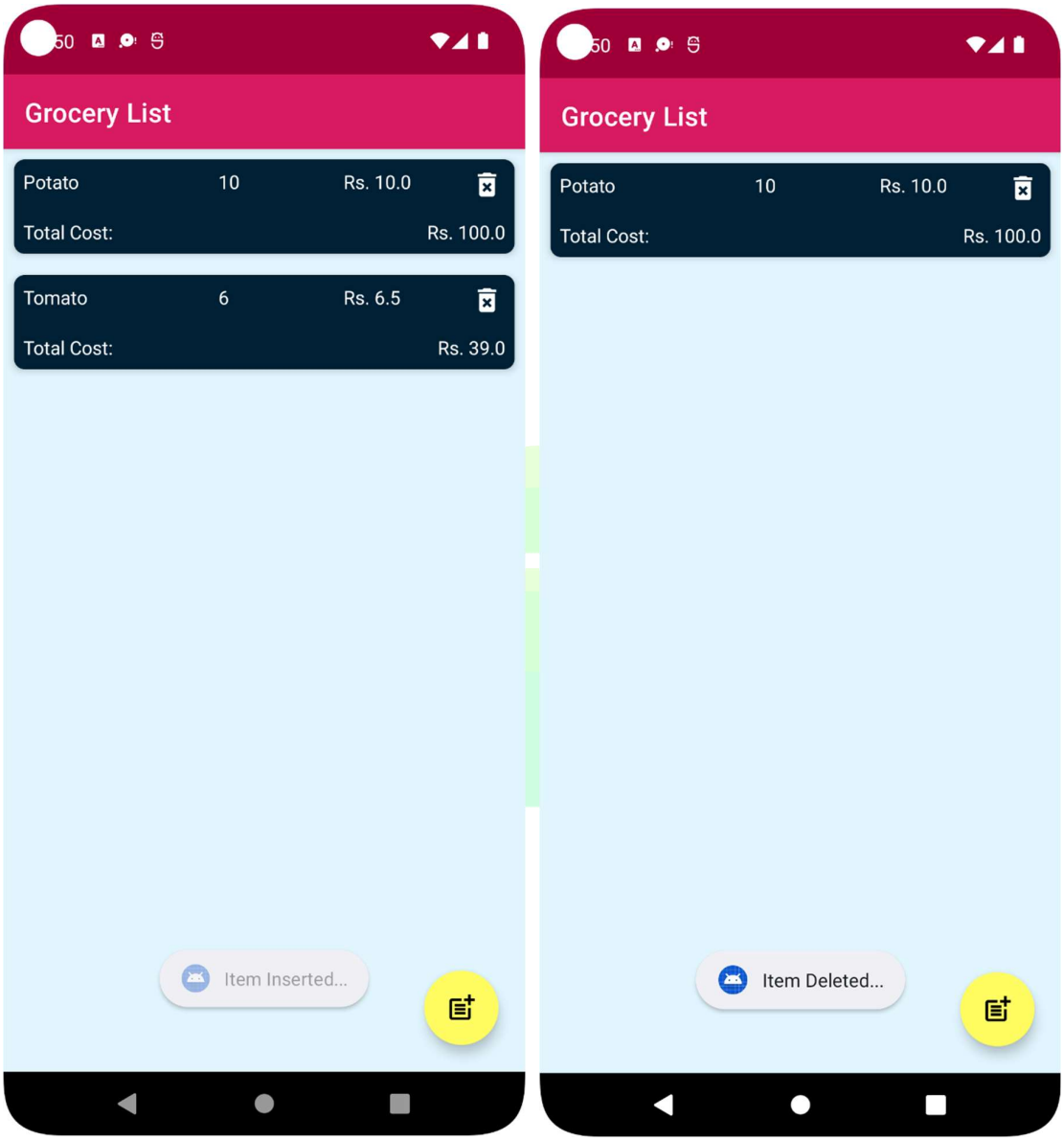
RESULT

We created a basic level solution for the problem. Below are some screenshots of the app.




Grocery List
Grocery List
Add Items to Cart
Enter Item Name
Enter Item Quantity
Enter Item Price
CANCEL
ADD


Grocery List
Add Items to Cart
Enter Item Name
Potato
Enter Item Quantity
10
Enter Item Price
10
CANCEL
ADD
Grocery List
Potato 10 Rs. 10.0
Total Cost: Rs. 100.0
Item Inserted...


Grocery List
Potato 10 Rs. 10.0
Total Cost: Rs. 100.0
Tomato 6 Rs. 6.5
Total Cost: Rs. 39.0
Item Inserted...
Grocery List
Potato 10 Rs. 10.0
Total Cost: Rs. 100.0
Item Deleted...

ADVANTAGES & DISADVANTAGES

- Advantages

    1. Easier to maintain item list compared in this form compared to pen-paper
    2. Easier to tally money spent on each item if we are buying multiple of it as it tells total cost of item for the quantity.

- Disadvantages

    1. Currently only support single user's single list.
    2. No authentication so anyone can access the list if they have access to list.
    3. No categorization or history of past data.

APPLICATIONS

Right now, this solution can be applied to solve daily listing problem it maintains the grocery inventory.

CONCLUSION

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.

FUTURE SCOPE

This application helps to store the list of items by Admin. In future we can also add scheduled addition of items according to requirement of user.

The Features are:

- Add User Panel
- Add Admin Panel
- Provide Login Authentication
- Add Image to user Product and Rating
- Provide data syncing between devices
- History to analyse spending habits
- Search and finding price for items added on the internet to suggest lower prices and saving money.
- Note field to add some instructions to be followed while shopping.

BIBILOGRAPHY

[1] Reference Video: Build a Grocery Android App using MVVM and Room Database in Kotlin | GeeksforGeeks
https://www.youtube.com/watch?v=vdcLb_Y71Ic

[2] Reference: Android Development Manual developer.android.com

[3] Reference: Room Dependencies
https://developer.android.com/codelabs/android-room-with-a-view-kotlin#3

[4] Resources and Reference: Material Design https://material.io/

[5] Icon Resource: Icons8 https://icons8.com/icons/set/grocery

APPENDIX

A. Source Code

Github Repo: https://github.com/smartinternz02/SPSGP-89987-Virtual-Internship---Android-Application-Development-Using-Kotlin

Email ID: kanaishkgarg@gmail.com

Github Profile: https://github.com/kanaishk

Demo Link: https://youtu.be/8LQLkdQzy2U

B. 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 sincere thanks to all of them.

I am highly indebted to **SmartInternz** (Experiential Learning & Remote Externship Platform to bring academia & industry very close for a common goal of talent creation) for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude towards member of (SmartInternz) for their kind co-operation and encouragement which help me in completion of this project.

My thanks and appreciations also go to people who have willingly helped me out with their abilities as well as my fellow learners that helped me found solution to problems that I was not able to solve myself.

C. Reflection Notes

I thoroughly enjoyed my internship and had a very valuable experience under my belt. I know this will help when looking for jobs and needing references.

I know that practical experience is the best, and internships give students that hands-on experience they need. I feel that quality internships are essential to develop key skills that we can't get in a classroom. Skills such as multitasking, communicating, learning to deal with diversity, and dealing with deadlines are different when you are working for someone else, not yourself like everyone do it college. Internships are also a great way to network with people in the industry. Our mentor and co-workers were great about giving us contacts and referring us to open positions in the industry.

I have learned that stressing over little things will not get us anywhere. Another aspect that I learned throughout the internship is to never be afraid to ask lots of questions. By asking questions we get answers.