My Grocery (App Report)

1. Introduction

a. OverView

My Grocery is an android app supported on API 19 or above which help you to store list of grocery items with their price and quantity in an organized manner.

b. Purpose

By the use of this app the user can store their day to day grocery items that need to be purchased from shops in our app database. so, that they don't forget any item in future and have rough idea of overall budget required.

2. Literature Survey

2.1 Existing problem

We buy groceries more often going to market and if the no of items is big then we can not remember all of items that need to be purchased with their respective quantity. So, for that what we try to do is to make list in form of some physical paper. But it leads to wastage of papers unnecessary.Also, we need to calculate manually the overall cost of each item by multiplying quantity x Rate which can be time consuming and also error prone.

2.2 Proposed Solution-

Now, an ultimate solution to this problem is to have an app which can store all our items that need to be buy ed that where My Grocery comes in place. My Grocery is simple to use, reliable, User friendly listing app which keep records of your Grocery item with their Quantity, Rate and total cost made. So, you can have an esteemed budget before you go to shopping.

Functions:-

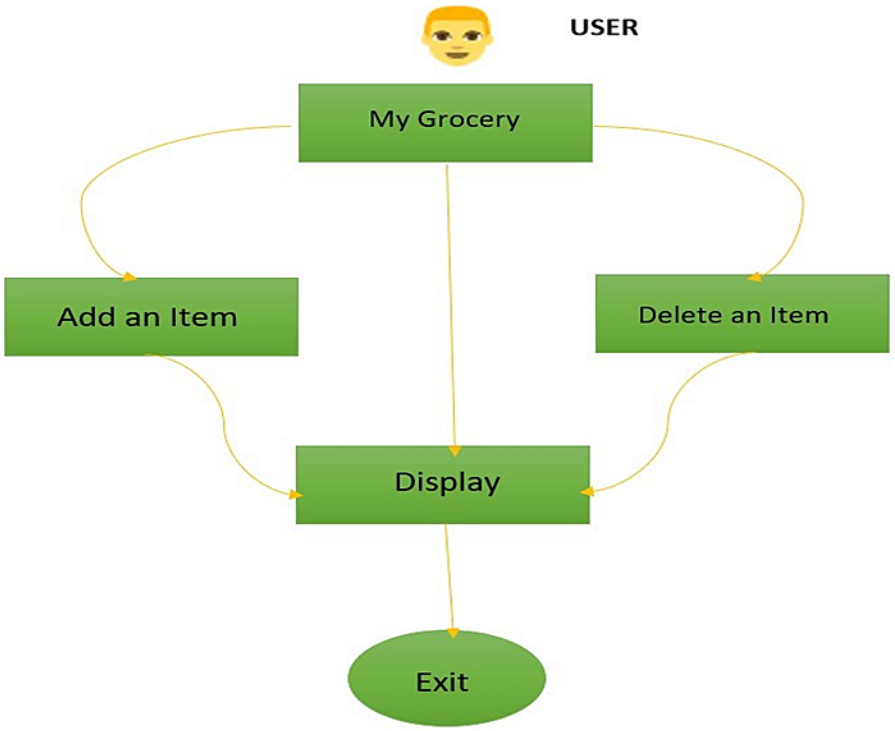
Add an item(Name, Quantity, Rate)

Delete an item

See total cost

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware/Software Requirements-

In order to run My Grocery app you must have an android device with good quality of touch sensitivity and have Android Version KitKat(API 19) or above with play store installed.

4. Experiment Investigation

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

LiveData: A data holder class that can be observed. Always holds/caches the latest version of data, and notifies 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 relevant lifecycle status changes while observing.

ViewModel: Acts as a communication center between the Repository (data) and the UI. The UI no longer needs to worry about the origin of the data. ViewModel instances survive Activity/Fragment recreation.

Repository: A class that you create that is primarily used to manage multiple data sources.

Entity: Annotated class that describes a database table when 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.

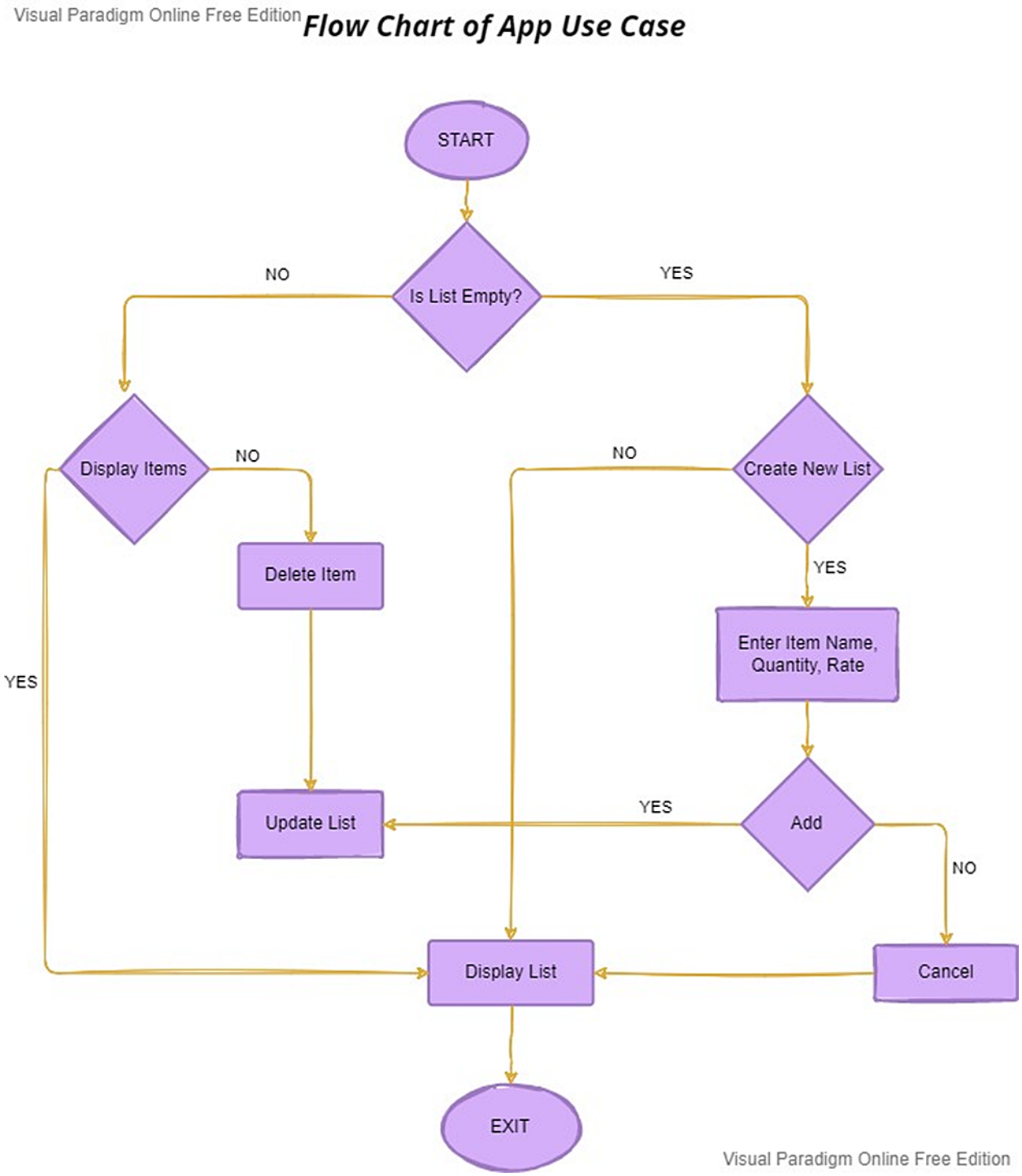
DAO: Data access object. A mapping of SQL queries to functions. When you use a DAO, you call the methods, and Room takes care of the rest.

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 number of views.

Coroutines: Coroutines are 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.

5. Flow Chart

a)



8. Advantages & Disadvantages

Adavantages-

1. Simple, convenient, clean UI so, that any group of user can use it.
2. See overall budget
3. Any no. of item can be added

Diadvantages-

4. We can't specify item quantity type(kg, l, m).
5. Item with long name appear to be break apart.
6. Redundancy also comes.

9. Conclusion

My Grocery is the simple data store app which uses MVVM architecture and Room library to store list of Grocery Items which helps the user to keep track of needed.

10. Future Scope

More functionalities can be added such that from same app user can place orders of items from nearest Grocery Center.

Ui can be made more attractive. Option to Increase or decrease quantity of item or update an existing item.

11. Proof of Project

Github Link: https://github.com/smartinternz02/SPSGP-75899-Virtual-Internship---Android-Application-Development-Using-Kotlin

Project Demo: https://www.youtube.com/watch?v=6JTX0Grt46M

Google Developer Profile: g.dev/sharves