Mind Grocery App Report

1. Introduction :

1.1Overview :

This is an android app that helps the user to make a list and add grocery items along with its price and quantity. And most importantly user can see cart price and selected cart price viewing single list or combined list.

1.2Purpose :

As we can't remember everything, users frequently forget to buy the things they want to buy. However, with the assistance of this app, you can make a list of the groceries you intend to buy so that you don't forget anything and also manage grocery expenditure and budget.

2. Literature Survey :

2.1 Existing Problem

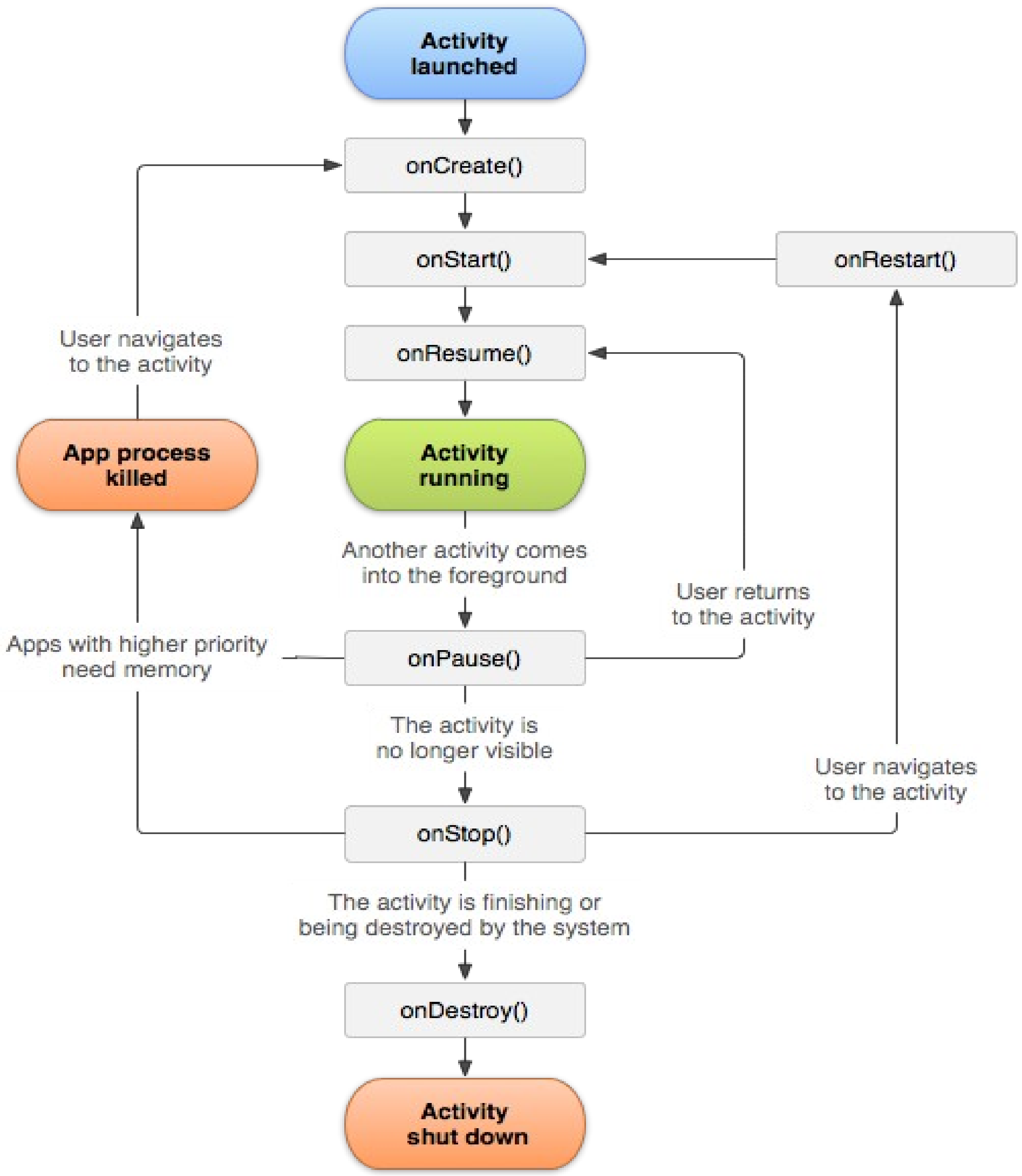
Users frequently forget items to buy because of which they have to run to shops again and again which is quite frustrating and waste of time.We end up wasting money on traveling which could be saved if we can keep a checklist and buy grocery at once with proper expenditure.

2.2 Proposed Solution

To overcome this problematic situation I built the Mind Grocery App which helps the user to make list/lists and manage their checklist and expenditure accordingly.

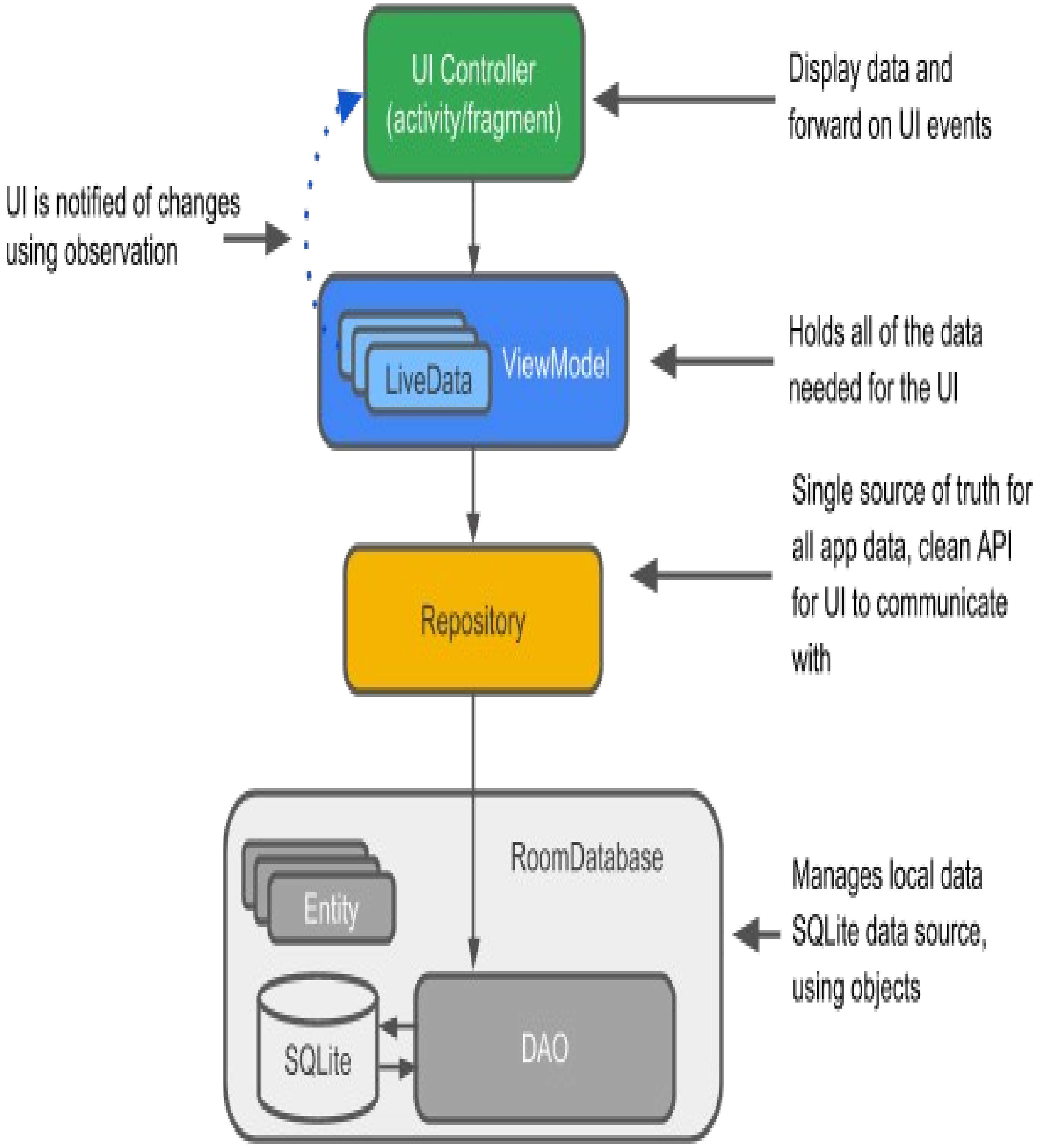
3. Theoretical Analysis:

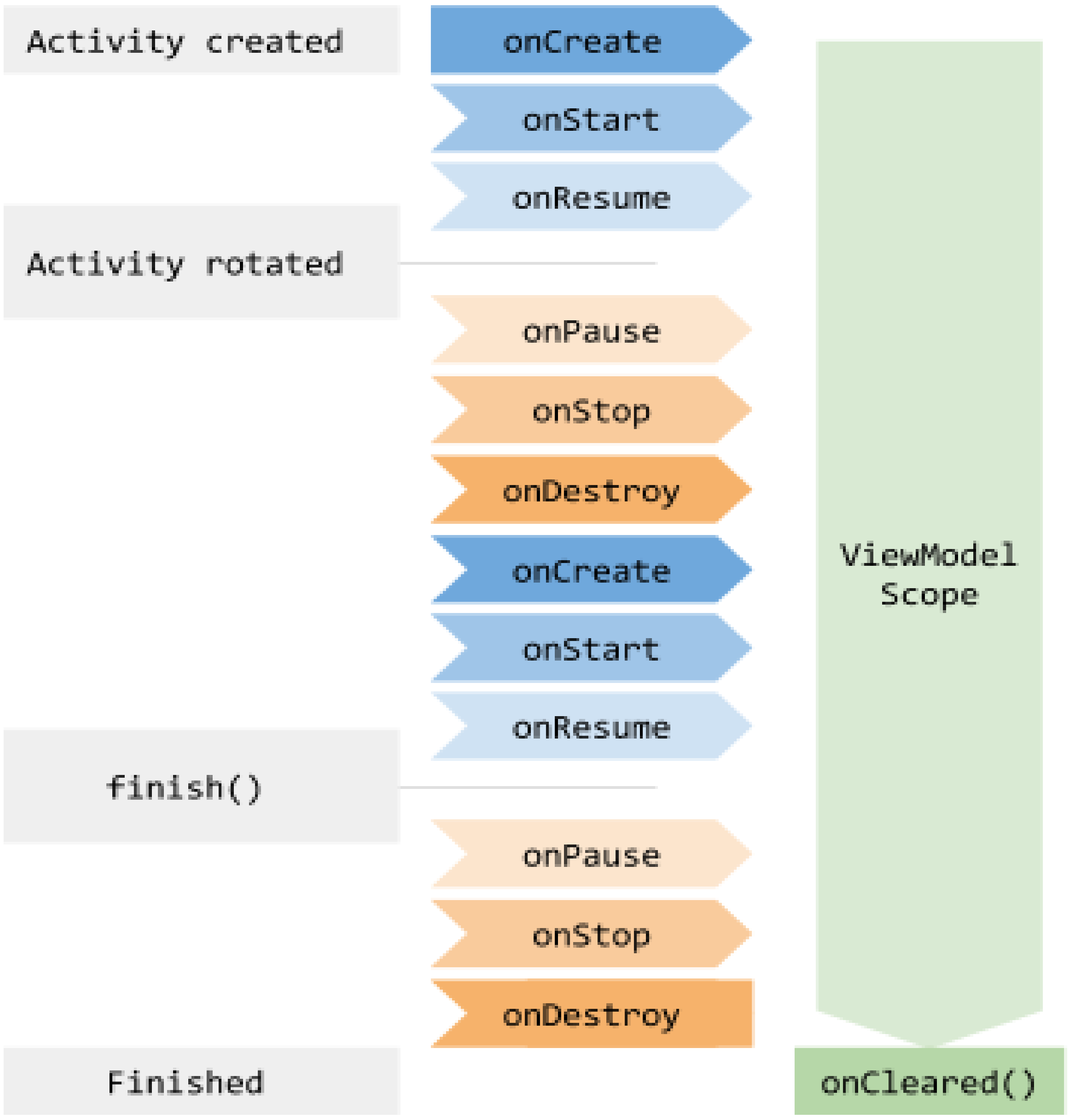
3.1 Block Diagram


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

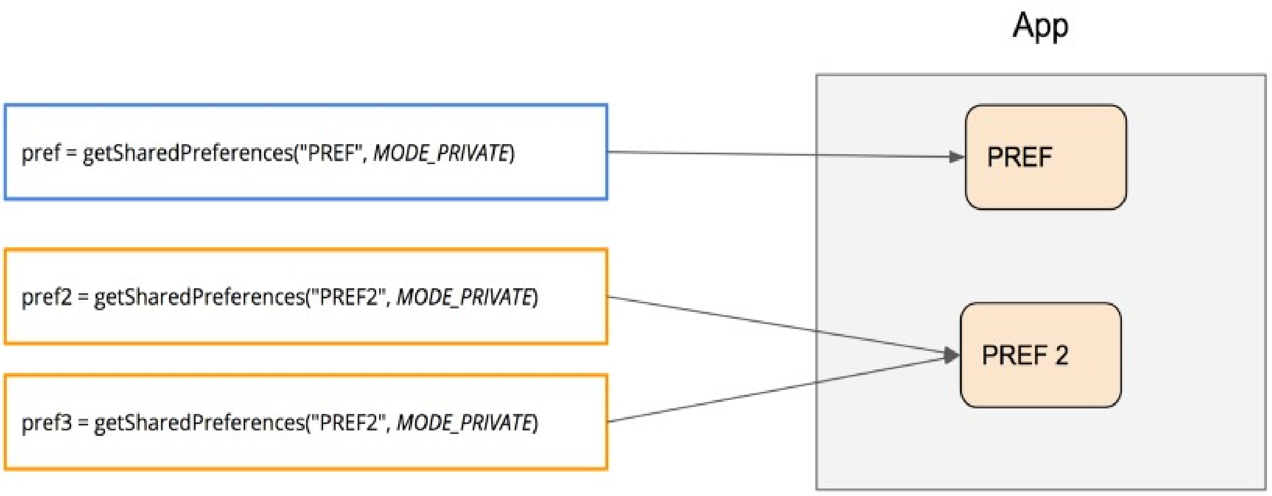
3

simplified illustration of the activity lifecycle.

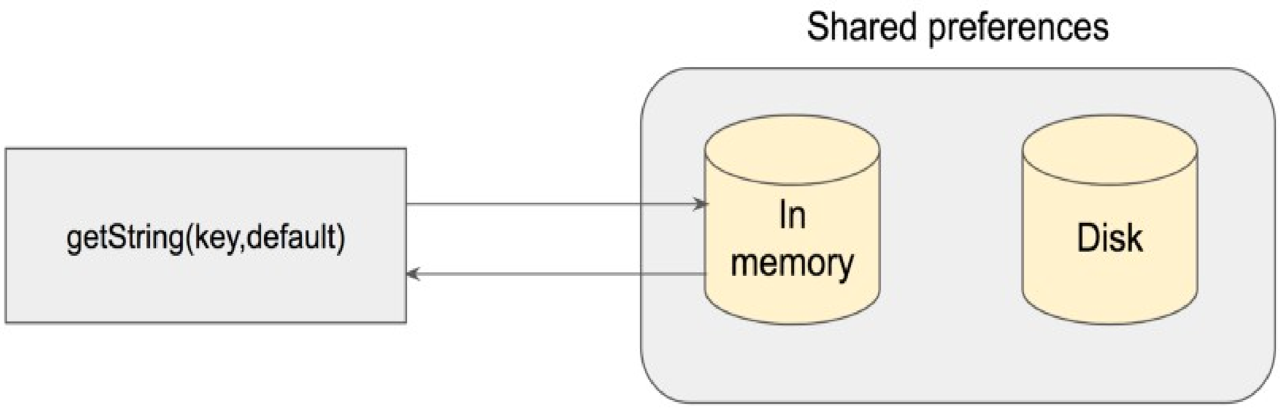

UI Controller (activity/fragment)
Display data and forward on UI events
UI is notified of changes using observation
LiveData
ViewModel
Holds all of the data needed for the UI
Repository
Single source of truth for all app data, clean API for UI to communicate with
RoomDatabase
Entity
SQLite
DAO
Manages local data SQLite data source, using objects



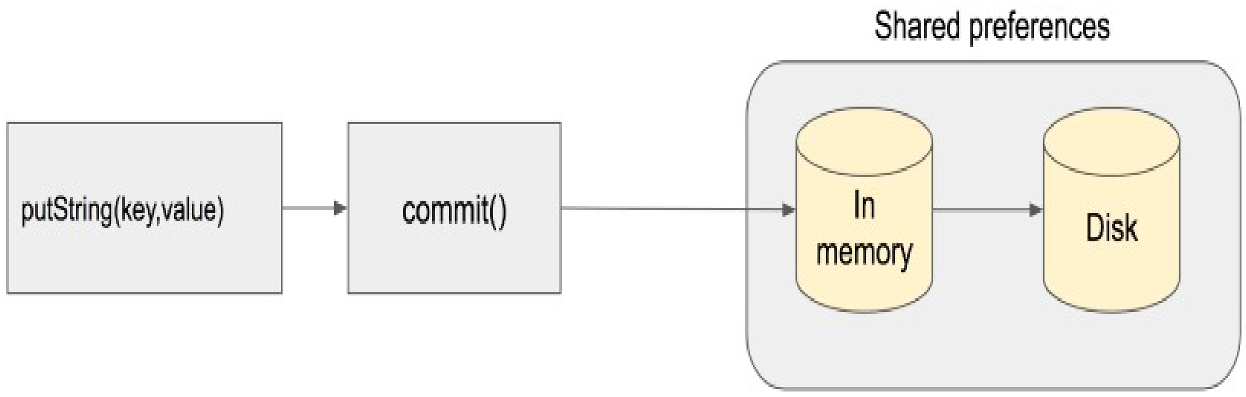
ViewModel Lifecycle


App
pref = getSharedPreferences("PREF", MODE_PRIVATE)
pref2 = getSharedPreferences("PREF2", MODE_PRIVATE)
pref3 = getSharedPreferences("PREF2", MODE_PRIVATE)
PREF
PREF 2

Notice that pref2 and pref3 uses the same instance because of the same name



Note that if it's the first time loading, it'll wait until all data is loaded to in-memory storage from disk.



Put a string to shared preferences by using commit

3.2Hardware/Software designing

● Windows 10 OS

An Android device or emulator

● Android Studio

The following required functionality is completed:

☑ User can persist grocery items and retrieve them properly on app restart.

☑ User can Add new grocery item with dialog box.

☑ User can Edit old grocery item with dialog box.

☑ User can delete an item by clicking on the delete button.

☑ User can successfully modify and remove items from the grocery list/cart.

The following optional features are implemented:

☑ User can create multiple grocery list.

☑ User can view items of respective grocery list in Grocery Screen.

☑ Read-Only Mode restricts edit and delete of items and the list itself.

☑ Budget Mode collects all grocery item from total lists present into one place as Combined Grocery List.

☑ About is briefly explained to the user with a Custom Dialog.

☑ Persist the grocery items into Room Database instead of a text file.

☑ Persist the setting into private text file.

☑ Improve style of the grocery items in the list using a custom adapter.

☑ Toast shows when the user try to save with no data filled.

☑ Check Box is provided in the item which helps user in two ways.

☑ Firstly, it marks the item as already dictated.

☑ Secondly, the sum of all checked items in cart is shown as the TOTAL CHECKED COST.

☑ The sum of all items in the cart is shown as the TOTAL CART COST.

☑ Combined Grocery list helps to manage items at one place which helps easy correction and ultimately manage user's grocery budget.

4. <u>Experimental Investigation:</u>

In this App, I have used MVVM pattern , Room for database, Coroutines and Recycler View to display the lists and their items. I have used shared preference for saving settings.
LiveData: A data holder class that can be observed. Always holds/caches the latet version of data, and notifies its observers when data has changed. LiveData is lifec
-ycle
aware. UI components just observe relevant data and don't stop or resume obser
vation.
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 t
he UI. The UI no longer needs to worry about the origin of the data. View
Model 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 t
he DAO to issue queries to the SQLite database.

Shared Preferences: it is the way in which one can store and retrieve small amounts of primitive data as key/value pairs to a file on the device storage such as String, int, float, Boolean that make up your preferences in an XML file inside the app on the device storage.

SQLite database: It is a open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation**.** On device storage. The Room persistence library creates and maintains this database for you.

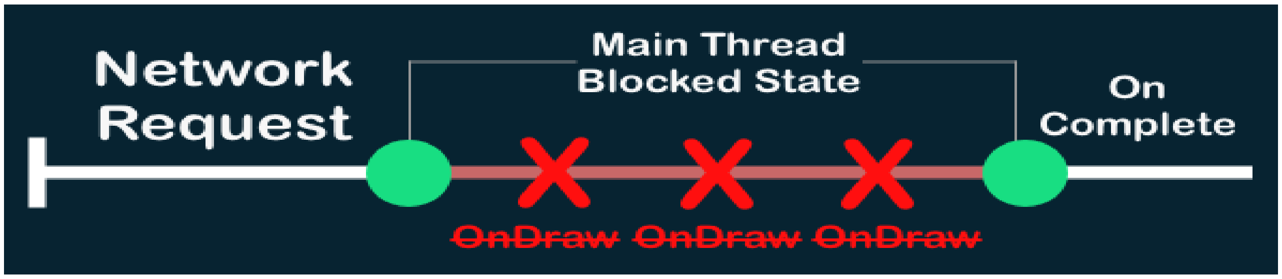
DAO: Data access object. A mapping of SQL queries to functions. When you use a DAO, you call the methods, and Room takes care of the rest.

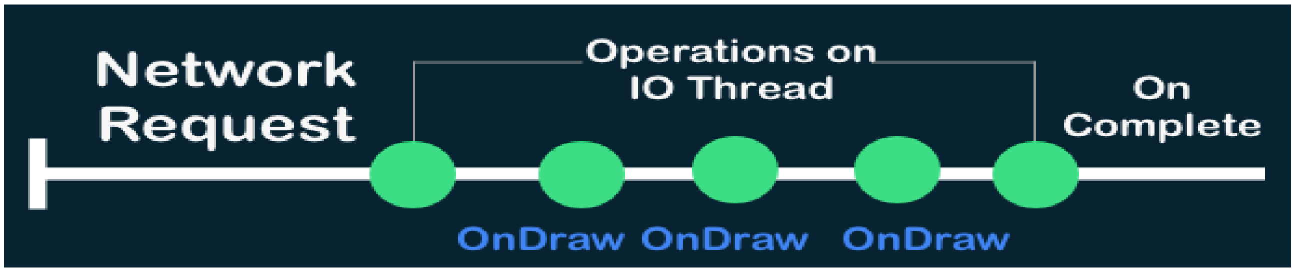
RecyclerView: It is a container and is used to display the collection of data in a lar
ge
amount of dataset that can be scrolled very effectively by maintaining a limited nu
mber of views.

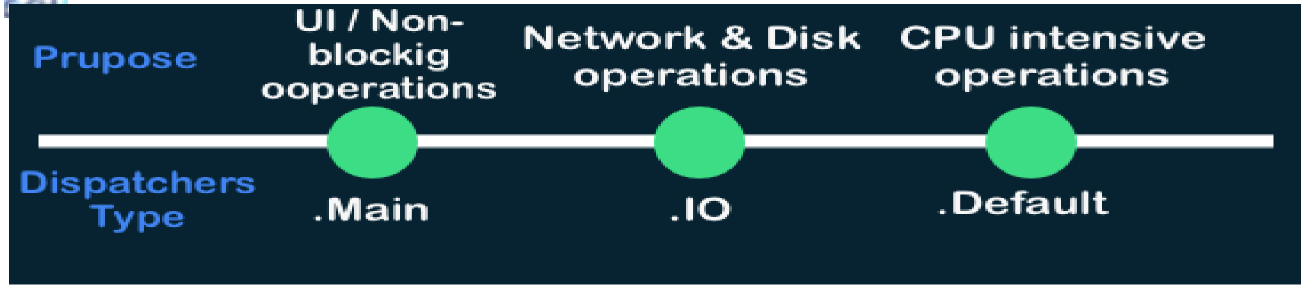
Coroutines: A coroutine is an instance of suspendable computation.

The main advantages of using Coroutines:

- They are **light-weight**
- Built-in **cancellation** support
- Lower chances for **memory leaks**
- **Jetpack** libraries provide coroutines support







Dispatchers.IO: The CoroutineDispatcher is designed for offloading blocking IO tasks to a shared pool of threads and networking operations. Additional threads in this pool are created and are shut down on demand

Room vs SQLite

Room is an ORM, Object Relational Mapping library. In other words, Room will map our database objects to Java objects. Room provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite.

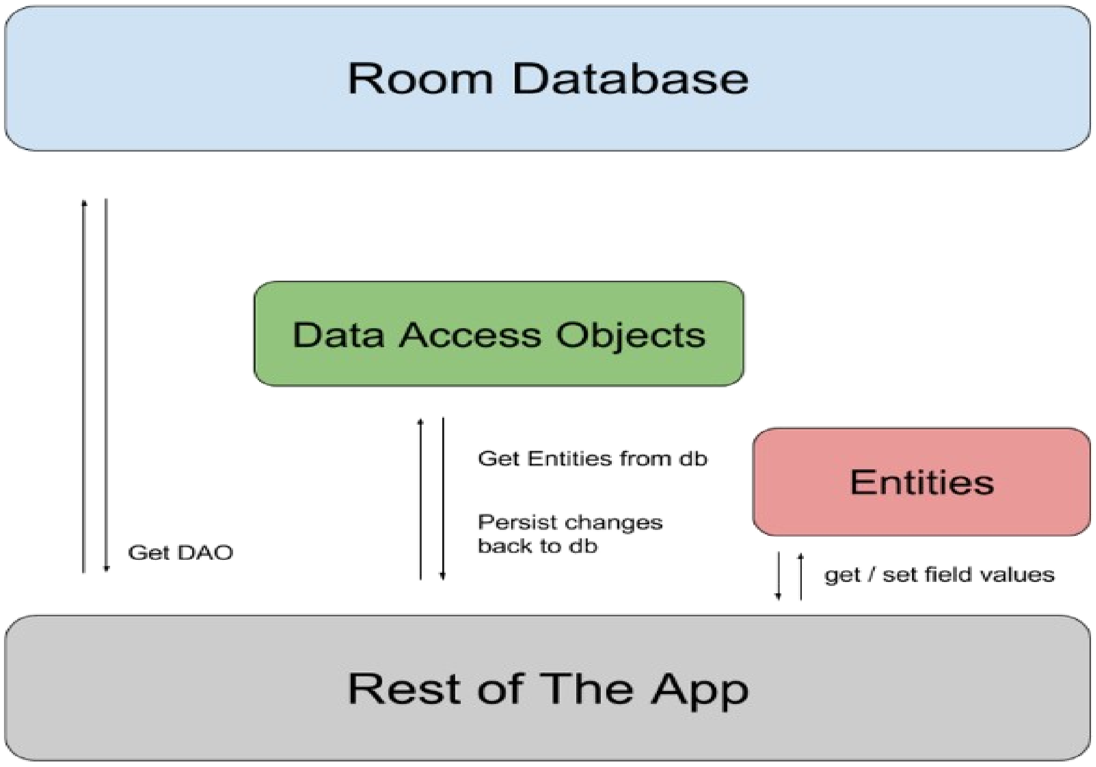
Difference between SQLite and Room persistence library:-

In the case of SQLite, There is no compile-time verification of raw SQLite queries. But in Room, there is SQL validation at compile time.

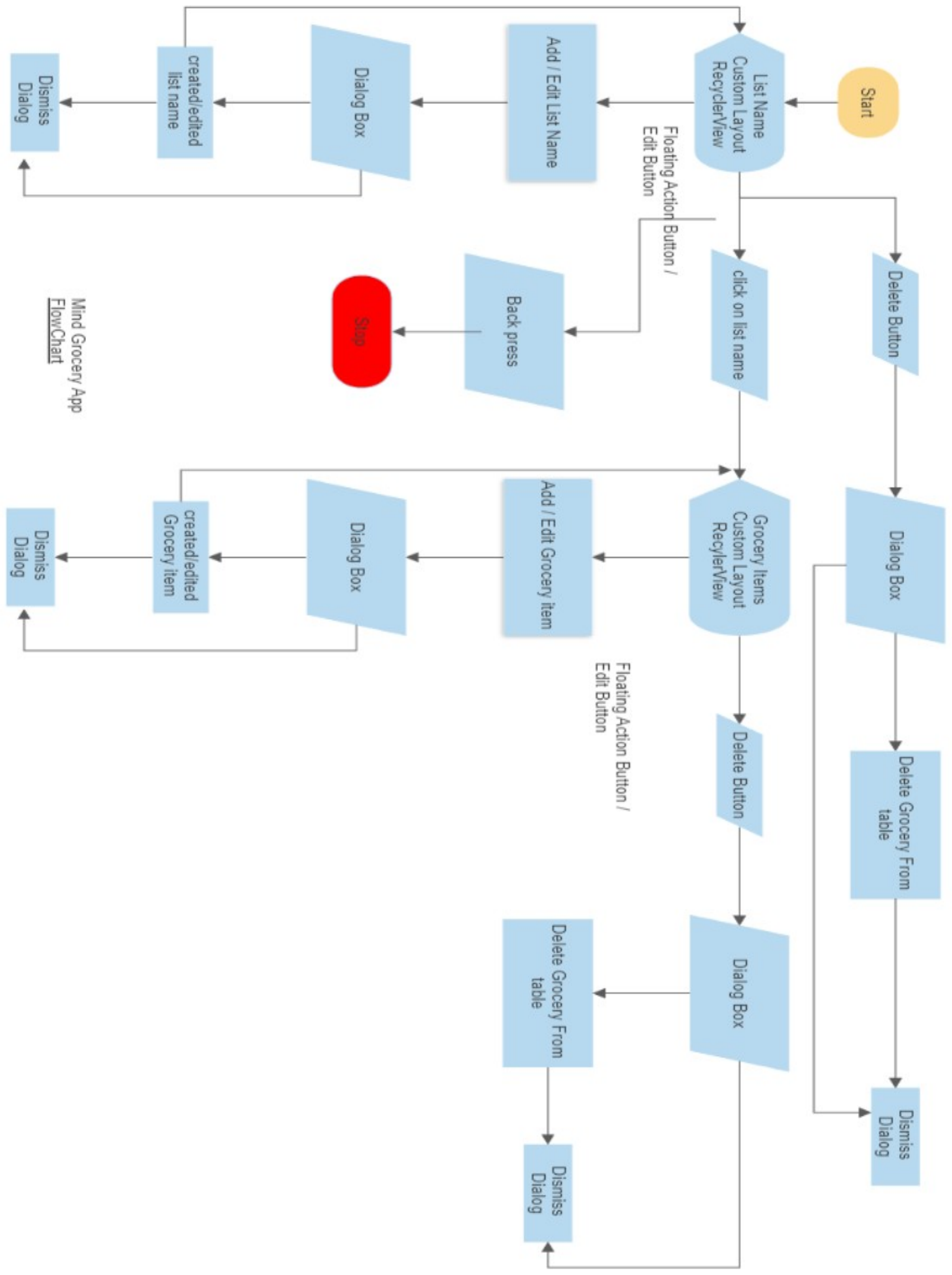
You need to use lots of boilerplate code to convert between SQL queries and Java data objects. But, Room maps our database objects to Java Object without boilerplate code.

As your schema changes, you need to update the affected SQL queries manually. Room solves this problem.

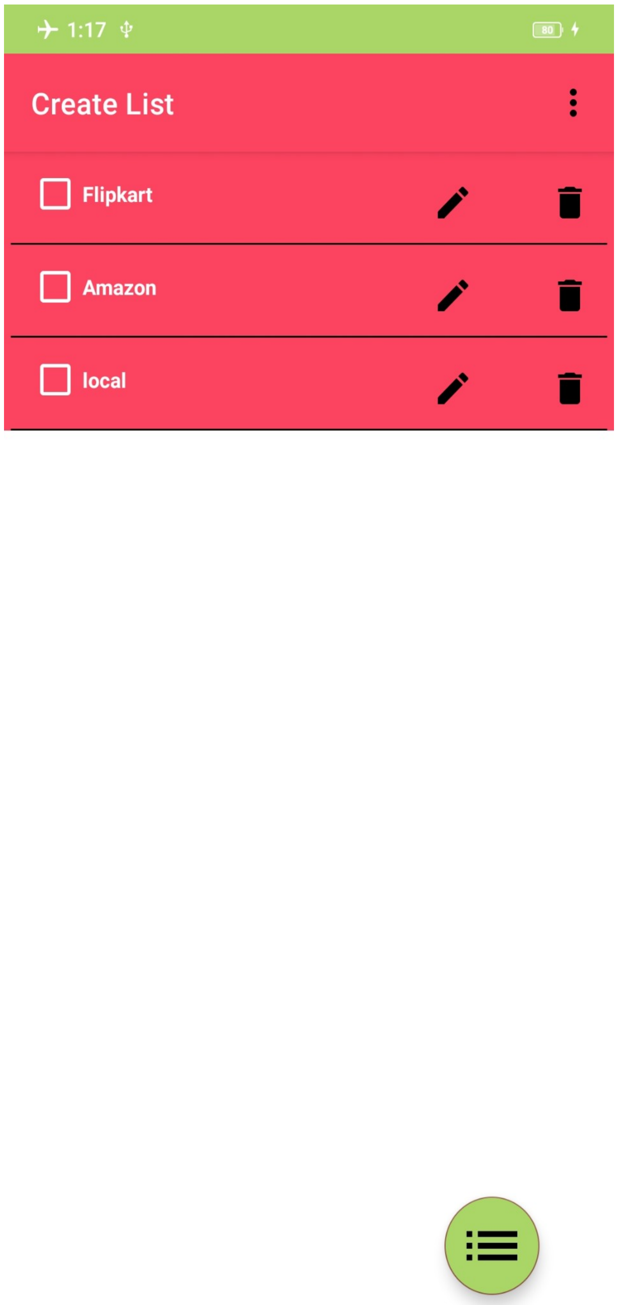
<u>Room is built to work with LiveData and RxJava for data observation, while SQLite does not.</u>



5.Flowchart:


Start
List Name Custom Layout RecyclerView
Add / Edit List Name
Floating Action Button / Edit Button
Dialog Box
created/edited list name
Dismiss Dialog
click on list name
Delete Button
Back press
Stop
Dialog Box
Delete Grocery From table
Dismiss Dialog
Grocery Items Custom Layout RecylerView
Add / Edit Grocery item
Floating Action Button / Edit Button
Dialog Box
created/edited Grocery item
Dismiss Dialog
Delete Button
Dialog Box
Delete Grocery From table
Dismiss Dialog
Mind Grocery App FlowChart

6. Result:

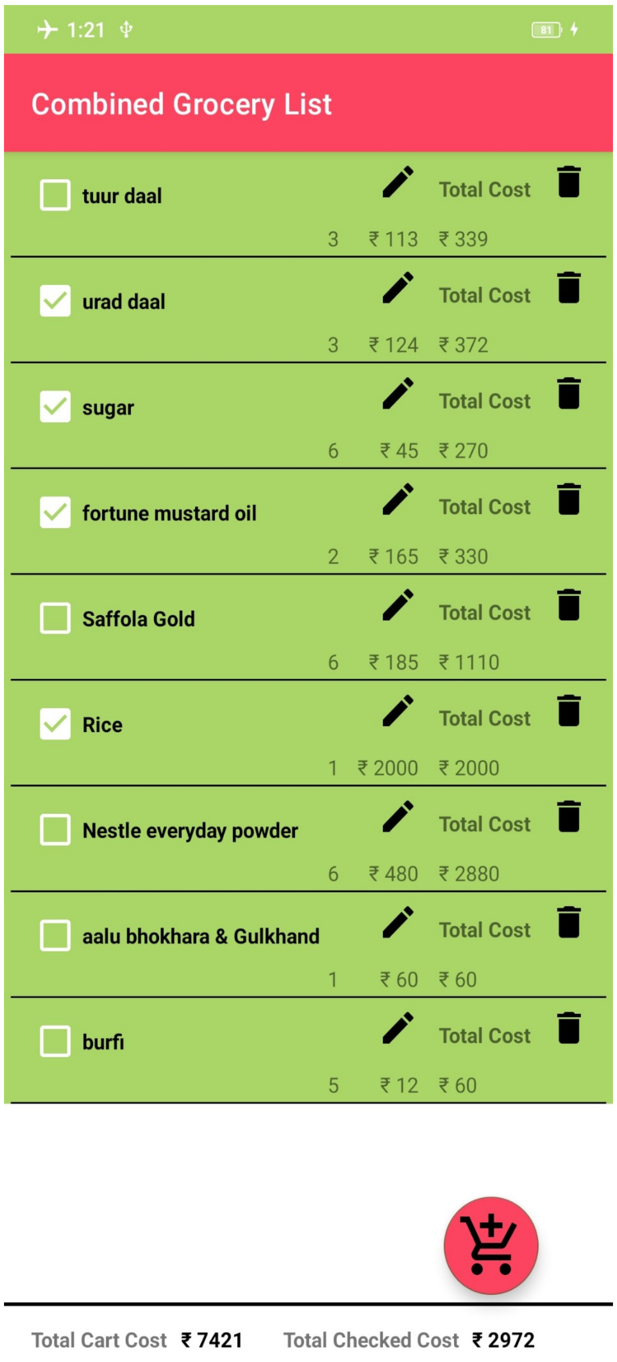


1:17

Create List

- [ ] Flipkart
- [ ] Amazon
- [ ] local

List Screen



1:21

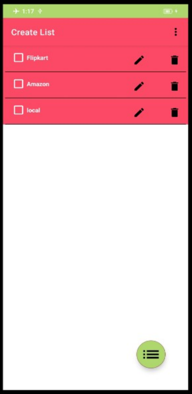
Combined Grocery List

☐	tuur daal	3	₹ 113	Total Cost ₹ 339
✓	urad daal	3	₹ 124	Total Cost ₹ 372
✓	sugar	6	₹ 45	Total Cost ₹ 270
✓	fortune mustard oil	2	₹ 165	Total Cost ₹ 330
☐	Saffola Gold	6	₹ 185	Total Cost ₹ 1110
✓	Rice	1	₹ 2000	Total Cost ₹ 2000
☐	Nestle everyday powder	6	₹ 480	Total Cost ₹ 2880
☐	aalu bhokhara & Gulkhand	1	₹ 60	Total Cost ₹ 60
☐	burfi	5	₹ 12	Total Cost ₹ 60

Total Cart Cost **₹ 7421** Total Checked Cost **₹ 2972**

Grocery Screen

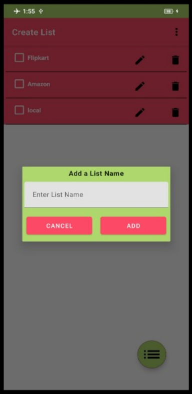
List Screen


Create List

☐ Flipkart

☐ Amazon

☐ local


Create List

☐ Flipkart

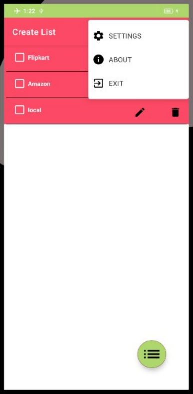
☐ Amazon

☐ local

Add a List Name

Enter List Name

CANCEL ADD


Create List

☐ Flipkart

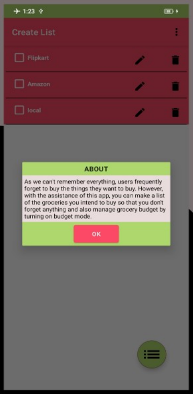
☐ Amazon

☐ local

SETTINGS

ABOUT

EXIT


Create List

☐ Flipkart

☐ Amazon

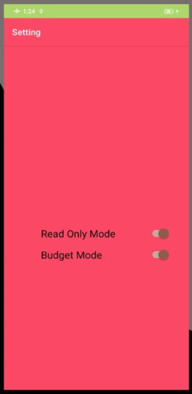
☐ local

ABOUT

As we can't remember everything, users frequently forget to buy the things they want to buy. However, with the assistance of this app, you can make a list of the groceries you intend to buy so that you don't forget anything and also manage grocery budget by turning on budget mode.

OK

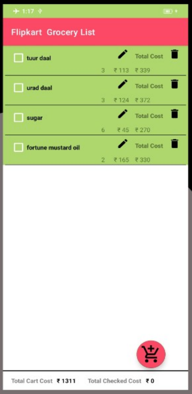
Setting Screen


Setting

Read Only Mode

Budget Mode

Grocery Screen


Flipkart Grocery List

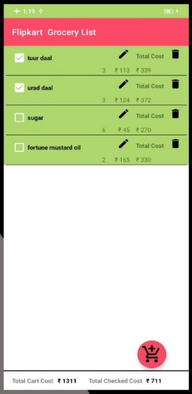
☐ tuur daal — Total Cost — 3 ₹ 113 ₹ 339

☐ urad daal — Total Cost — 3 ₹ 124 ₹ 372

☐ sugar — Total Cost — 6 ₹ 45 ₹ 270

☐ fortune mustard oil — Total Cost — 2 ₹ 165 ₹ 330

Total Cart Cost ₹ 1311 Total Checked Cost ₹ 0


Flipkart Grocery List

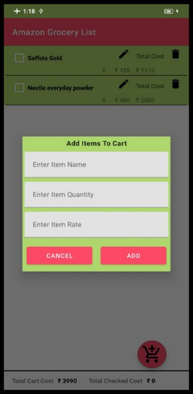
✓ tuur daal — Total Cost — 3 ₹ 113 ₹ 339

✓ urad daal — Total Cost — 3 ₹ 124 ₹ 372

☐ sugar — Total Cost — 6 ₹ 45 ₹ 270

☐ fortune mustard oil — Total Cost — 2 ₹ 165 ₹ 330

Total Cart Cost ₹ 1311 Total Checked Cost ₹ 711


Amazon Grocery List

☐ Saffola Gold — Total Cost — 6 ₹ 185 ₹ 1110

☐ Nestle everyday powder — Total Cost — 6 ₹ 480 ₹ 2880

Add Items To Cart

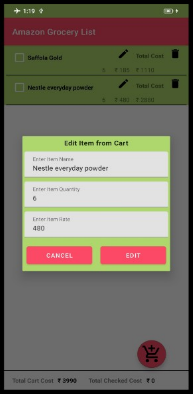
Enter Item Name

Enter Item Quantity

Enter Item Rate

CANCEL ADD

Total Cart Cost ₹ 3990 Total Checked Cost ₹ 0


Amazon Grocery List

☐ Saffola Gold — Total Cost — 6 ₹ 185 ₹ 1110

☐ Nestle everyday powder — Total Cost — 6 ₹ 480 ₹ 2880

Edit Item from Cart

Enter Item Name: Nestle everyday powder

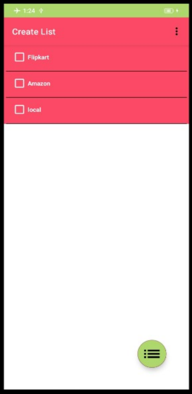
Enter Item Quantity: 6

Enter Item Rate: 480

CANCEL EDIT

Total Cart Cost ₹ 3990 Total Checked Cost ₹ 0

ReadOnly Mode


Create List

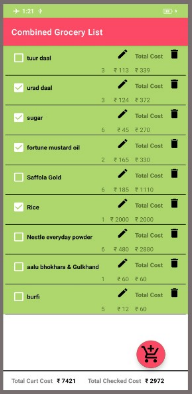
☐ Flipkart

☐ Amazon

☐ local

BUDGET MODE

ReadOnly Mode


Combined Grocery List

☐ tuur daal — Total Cost — 3 ₹ 113 ₹ 339

✓ urad daal — Total Cost — 3 ₹ 124 ₹ 372

✓ sugar — Total Cost — 6 ₹ 45 ₹ 270

✓ fortune mustard oil — Total Cost — 2 ₹ 165 ₹ 330

☐ Saffola Gold — Total Cost — 6 ₹ 185 ₹ 1110

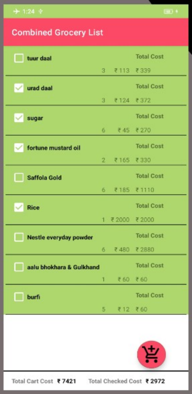
✓ Rice — Total Cost — 1 ₹ 2000 ₹ 2000

☐ Nestle everyday powder — Total Cost — 6 ₹ 480 ₹ 2880

☐ aalu bhokhara & Gulkhand — Total Cost — 1 ₹ 60 ₹ 60

☐ burfi — Total Cost — 5 ₹ 12 ₹ 60

Total Cart Cost ₹ 7421 Total Checked Cost ₹ 2972


Combined Grocery List

☐ tuur daal — Total Cost — 3 ₹ 113 ₹ 339

✓ urad daal — Total Cost — 3 ₹ 124 ₹ 372

✓ sugar — Total Cost — 6 ₹ 45 ₹ 270

✓ fortune mustard oil — Total Cost — 2 ₹ 165 ₹ 330

☐ Saffola Gold — Total Cost — 6 ₹ 185 ₹ 1110

✓ Rice — Total Cost — 1 ₹ 2000 ₹ 2000

☐ Nestle everyday powder — Total Cost — 6 ₹ 480 ₹ 2880

☐ aalu bhokhara & Gulkhand — Total Cost — 1 ₹ 60 ₹ 60

☐ burfi — Total Cost — 5 ₹ 12 ₹ 60

Total Cart Cost ₹ 7421 Total Checked Cost ₹ 2972

Mind Grocery App

BY ASIK ALI KHAN

7.Conclusion:

I applied the concepts such as Room Database, Coroutines, Shared preference and MVVM architecture pattern into this project.

Working on this project made me confident enough to apply my knowledge into real life project.

The project can be upgraded to have share pdf functionality and invoice making functionality so it can be useful to the small retailers.

Submitted by: ASIK ALI KHAN
SBID : SB20220211572
Application id – SPS_APL_20220072278

SmartInternz GitHub Repository Submitted link

Video Walk Through

My Developer Profile

8.Reference:

● Google: https://www.google.com/

● Geeksforgeeks: https://www.geeksforgeeks.org/how-to-build-a-grocery-androidapp-using-mvvm-and-room-database/

● Android Developer: https://developer.android.com/codelabs/android-room-witha-view-kotlin#0

● YouTube: https://www.youtube.com/watch?v=vdcLb_Y71Ic

● SmartInternz: Student Dashboard (smartinternz.com)

9. Appendix:9.1 Source Code:

Note:- Since the page limit is exceeding I can't put the source code here .Please check the GitHub Repository for the complete source code of the project.

My GitHub Repository