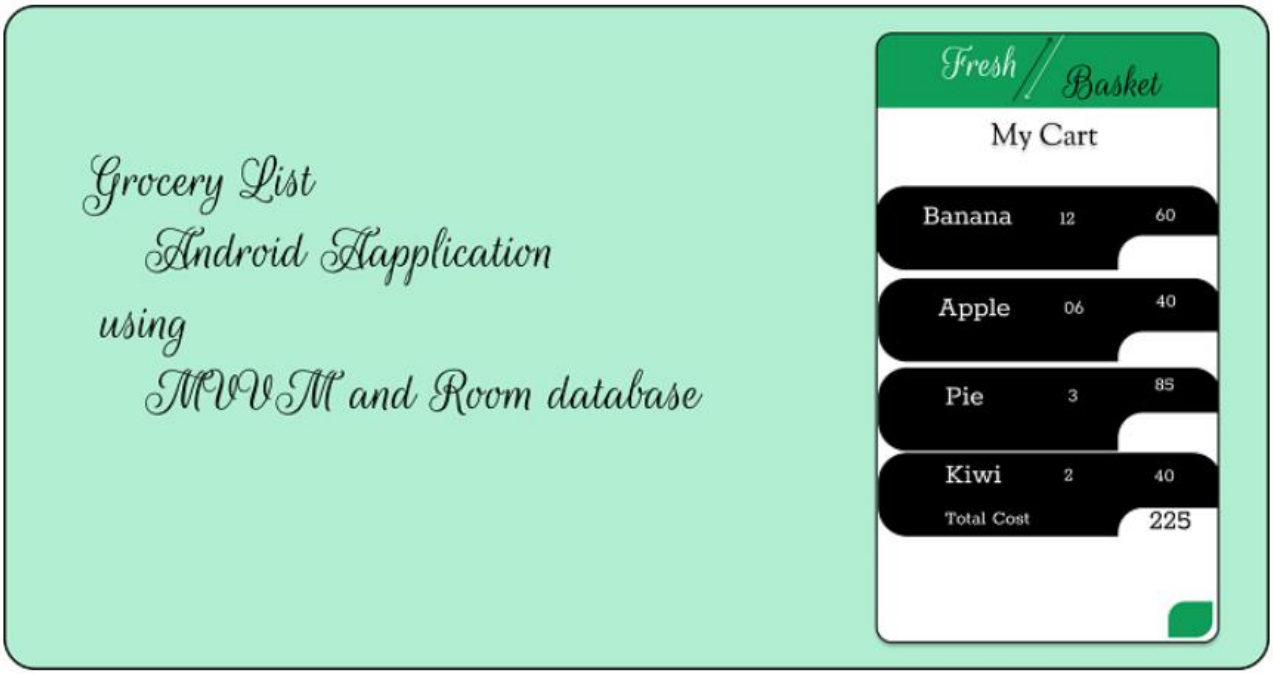
VIRTUAL INTERNSHIP - ANDROID APPLICATION DEVELOPMENT USING KOTLIN
GROCERY APPLICATION PROJECT

INTRODUCTION

Overview

In the recent decade, electronic commerce and online retailing have unquestionably happened to be essential components of the global retail landscape. Such as various other businesses, the retail scene has changed dramatically with the advent of the internet. The number of digital buyers proliferates worldwide as internet access and adoption rise rapidly, leading to online shopping increasing year after year

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, we can note down your grocery items that we are going to purchase, by doing this we can't forget any items that we want to purchase. A sample image is given below to get an idea about what we are going to do in this Project. Note that we are going to implement this project using the **Kotlin language**.



We're going to discuss how to create a Grocery Android App using MVVM and Room Database in Kotlin. With this application, the user will be able to note down the grocery items that he/she is going to purchase. 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. So, let's get started now.

PURPOSE AND USE OF THIS PROJECT :

Purpose of this project is to create a Grocery Android App using MVVM and Room Database in Kotlin. So with the help of this app, you can note down your grocery items that we are going to purchase. We also going to learn how to build different apps using new technologies like kotlin and how to use platforms like Android Studios.

We will also learn about **MVVM (Model View ViewModel).**

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. Before jumping to the project let's understand these terms.

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.

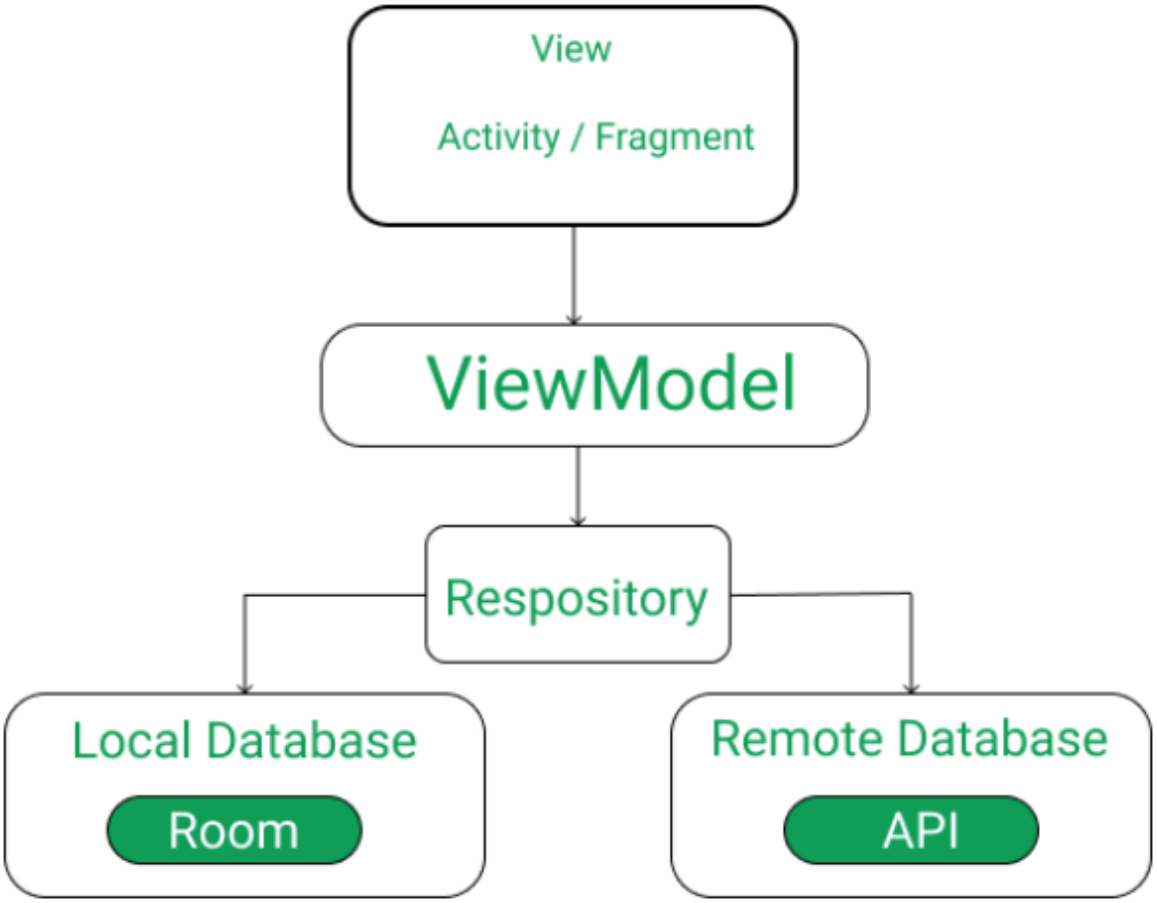
As we all know that the grocery industry is enormous and is increasing regardless of the challenges it is facing. Everyone needs provisions for their home and visits the grocery stores almost every month. From the past half-decade, the trend of **Online Grocery Market in the India** has become more popular with the increase in Grocery App solutions as customers find it so convenient to order from the Grocery app to go and get the groceries by themselves. Due to the busy schedules for everyone these days and also many other reasons to look for easy ways to get the work done. With an increase in technology, people are also getting busy. Also, there is a lot of population all these made grocery stores to shift to an online buying app which has brought some challenges. Still, are you in dilemma whether to **develop a Grocery App solution for your Grocery Business before** you need to overcome a few problems.

LITERATURE SURVEY

- **Low-profit margins-eating away profits:** Customers always prefer **Online Grocery app solution** as they expect good deals with less price. But after spending so much on shifting the store to online grocery app development but it is a too tough task to keep less cost for the products and sell them to customers which will ultimately bring loss to the company. Also, many people would like to test the vegetables physically rather than seeing them on the screen, which is impossible in online.
    - **Solution:** By including non-perishable items like packaged foods and personal care item etc. whose storage and delivery cost are less, you can save some expenses as well as satisfy the customer. You can merge the offline and online business where people can order online and collect them whenever possible from the store. This will minimise the operation cost to some extent.
- **Disorganised and Inefficient Delivery System:** Customers will never plan before they order from a grocery store, they order it when they like to eat something or need any fruits and vegetables at that moment and expect it to reach them without any delay. The customers who order from online grocery app solution will compromise with the spontaneous buying experience. Many people still prefer to visit the store rather than ordering things online. As most of the people are working, they are not available to collect their stuff when it is delivered home.
    - **Solution:** This can be solved only if the companies follow some delivery schedules. A buyer should be able to receive his order within 30 mins, which must be implemented by **Online grocery app development.** By this, the buyer can get a clear picture of the delivery so that they can manage their time.
- **Deep-rooted consumer behaviour:** One more significant challenge faced by the online grocery app solution is that people do not want to order online as they are habituated to go to the store and pick up the stuff. There is this belief that the things that are sold online are not fresh and tasty. As per the recent survey, it is found that 4% of the customers in Los Angels and 16% in New York bought their grocery online and remaining people still prefer to use the traditional ways to get their groceries.
    - **Solution:** Customers have shown more interest in organic products, but then this trend cannot sustain as there are insufficient biological resources. You can find out different ways of how customers can get those natural products at a reasonable price. Another thing about the customer is that they don't only see the quality of the food, but also they are concerned about the brand value of your store. Having a good brand recognisation attracts more customers to buy from your store.
- **Difficulty in penetration in small towns:** People in the small cities do not have much means of entertainment and for them going to store is the means of engaging with other people. On the other hand, most of the people who buy the stuff directly are habituated in bargaining, which is not possible when ordering through online. Some people are not aware of technology and are not willing to use it as the task of downloading the app and order seems like a big deal for them.
    - **Solution:** The grocery store owners must plan to conduct some awareness programs to such people and grow your brand reputation. By this, the people will show interest to go for online ordering from your grocery app development.

THEORITICAL ANALYSIS

Block diagram: Diagrammatic overview of the project.



After seeing this image, its understood how it will work. Let talk about the operations and concepts we gonna use in order to shape this project.

ROOM DATABASE

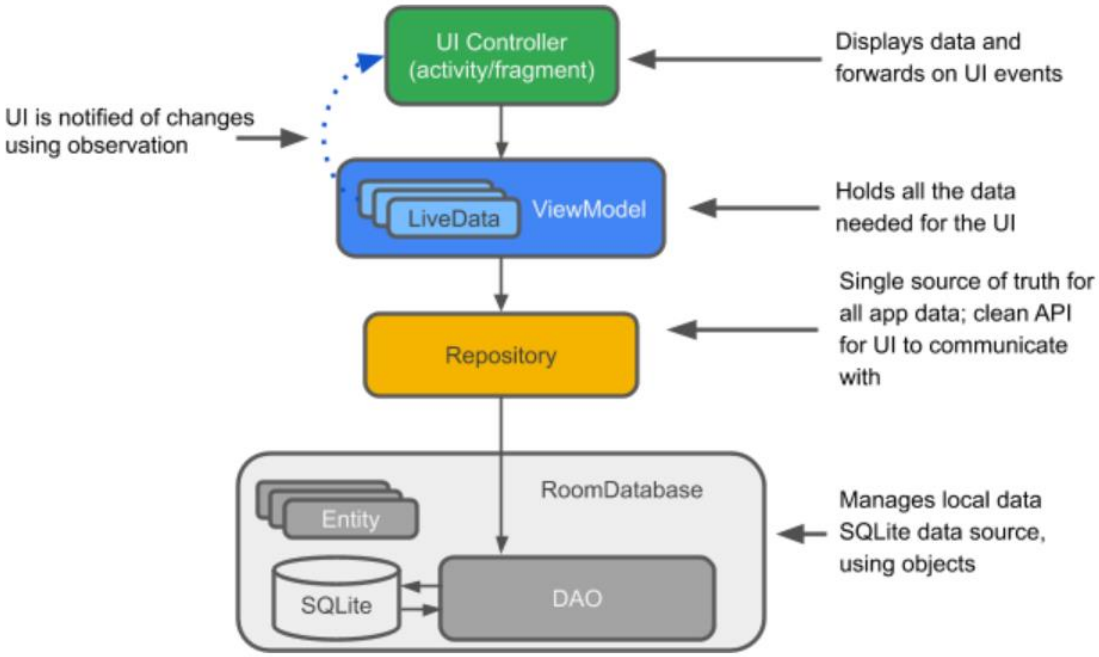
Room persistence library is a database management library and it is used to store the data of apps like grocery item name, grocery item quantity, and grocery item price. Room is a cover layer on SQLite which helps to perform the operation on the database easily.

RECYCLEVIEW

Recycler View 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.



HARDWARE / SOFTWARE DESIGNING HARDWARE AND SOFTWARE REQUIREMENTS OF THE PROJECT.

HARDWARE USED:

1. Laptop or PC with Android studio installed into it along with a good internet connection.
2. Android mobile phone for running our app(if not we can use Emulator as well.)

SOFTWARE AND TECHNOLOGY REQUIREMENTS:

1. Andorid studio.
2. Github.
3. Emulator. etc
4. Kotlin.

EXPERIMENTAL INVESTIGATIONS:

Being a shopper, I enjoy shopping in the local grocery stores and noticed some problems faced by the common people while shopping. I noticed different age groups faced different types of problems while shopping. Some don't have much time, some of them are not getting offers, many of them forgot some of the items on the way and much more. I did a survey and highlighted some pain points down below.

I thought why not try to solve the problems with Grocery App. While designing a whole new app to serve such a wide variety of people sounds like an overwhelming task, conducting user research served as an effective way to narrow down on what is truly important to the people who would be using the product.

DEFINING THE GOALS

I started to think if out there, users are experiencing the same problem. This curiosity inspired me to validate my assumption, and ultimately produce a solution to solve the problem.

TARGET AUDIENCE

The target audience of the project is the people aged between **18–45 who do Grocery shopping on a daily basis.** Most people of this age **use smartphones** and have knowledge of Online Shopping.

RESEARCH

In order to obtain useful information about how the app should be designed, both surveys and non-direct interviews were conducted.
Five non-direct interviews were completed in which a casual conversation was had with the target demographic. Since it was difficult to find stay-at-home parents, we decided to ask five parents in general. The results were extremely beneficial in understanding both the shopping habits and the tendency to try the new online app. A few very important key points were derived from these interviews that would play a crucial role in helping to shape the app:

PEOPLE ENJOY

All have different lifestyles and life goals and scenarios.
Some of them enjoy shopping, meeting friends in their spare time and some felt it hectic due to their busy schedules and personal problems.
They also get fresh fruits and vegetables from the market. Being regular shoppers, they are also benefited from many deals.

FRUSTRATIONS

They don’t have much time to shop and spend quality time choosing fresh fruits and vegetables.
It’s difficult to find the location of the items that consumed a lot of time of people having busy schedules.
Household ladies don’t have online payment availability so they can’t get the cashback deals.
It’s difficult to know what’s on sale or about to be on sale.
It’s difficult to keep track of how much the items in my cart cost before checking out at the counter. Sometimes they also forget the items to buy.

Source Code:

MainActivity.kt:

```
package com.example.groceryapplication
import android.app.Dialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.lifecycle.Observer
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import
com.google.android.material.floatingactionbutton.FloatingActionButton
class MainActivity : AppCompatActivity(),
GroceryRVAdapter.GroceryItemClickInterface {
    lateinit var itemsRV: RecyclerView
    lateinit var addFAB: FloatingActionButton
    lateinit var list: List<GroceryItems>
    lateinit var groceryRVAdapter: GroceryRVAdapter
    lateinit var groceryViewModal: GroceryViewModal
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        itemsRV = findViewById(R.id.idRVItems)
        addFAB = findViewById(R.id.idFABAdd)
        list = ArrayList<GroceryItems>()
        groceryRVAdapter = GroceryRVAdapter(list, this)
        itemsRV.layoutManager = LinearLayoutManager(this)
        itemsRV.adapter = groceryRVAdapter
        val groceryRepository = GroceryRepository(GroceryDatabase(this))
        val factory = GroceryViewModalFactory(groceryRepository)
        groceryViewModal = ViewModelProvider(this,
factory).get(GroceryViewModal::class.java)
        groceryViewModal.getAllGroceryItems().observe(this, Observer {
            groceryRVAdapter.list = it
            groceryRVAdapter.notifyDataSetChanged()
        })
```

```
        addFAB.setOnClickListener {
            openDialog()
        }
    }
    fun openDialog() {
        val dialog = Dialog(this)
        dialog.setContentView(R.layout.grocery_add_dialog)
        val cancelBtn = dialog.findViewById<Button>(R.id.idBtnCancel)
        val addBtn = dialog.findViewById<Button>(R.id.idBtnAdd)
        val itemEdt = dialog.findViewById<EditText>(R.id.idEdtItemName)
        val itemPriceEdt =
dialog.findViewById<EditText>(R.id.idEdtItemPrice)
        val itemQuantityEdt =
dialog.findViewById<EditText>(R.id.idEdtItemQuantity)
        cancelBtn.setOnClickListener {
            dialog.dismiss()
        }
        addBtn.setOnClickListener {
            val itemName: String = itemEdt.text.toString()
            val itemPrice: String = itemPriceEdt.text.toString()
            val itemQuantity: String = itemQuantityEdt.text.toString()
            val qty: Int = itemQuantity.toInt()
            val pr: Int = itemPrice.toInt()
            if (itemName.isNotEmpty() && itemPrice.isNotEmpty() &&
itemQuantity.isNotEmpty()) {
                val items = GroceryItems(itemName, qty, pr)
                groceryViewModal.insert(items)
                Toast.makeText(applicationContext, "Item Inserted..",
Toast.LENGTH_SHORT).show()
                groceryRVAdapter.notifyDataSetChanged()
                dialog.dismiss()
            } else {
                Toast.makeText(applicationContext, "Please Enter all the
data..", Toast.LENGTH_SHORT).show()

            }
        }
        dialog.show()
    }
    override fun onItemClick(groceryItems: GroceryItems) {
        groceryViewModal.delete(groceryItems)
        groceryRVAdapter.notifyDataSetChanged()
        Toast.makeText(applicationContext, "Item Deleted..",
Toast.LENGTH_SHORT).show()
    }
}
```

Activit_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="@color/lavender"
    tools:context=".MainActivity">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/idRVItems"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/grocery_rv_item" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/idFABAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_margin="20dp"
        android:background="@color/mint"
        android:layout_alignParentRight="true"
        android:src="@drawable/ic_add"
        app:tint="@color/purple_700" />
</RelativeLayout>
```

GroceryDatabase.kt:

```
package com.example.groceryapplication
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 {
                instance = it
            }
        }
        private fun createDatabase(context: Context) =
            Room.databaseBuilder(
                context.applicationContext,
                GroceryDatabase::class.java,
                "Grocery.db"
            ).build()
    }
}
```

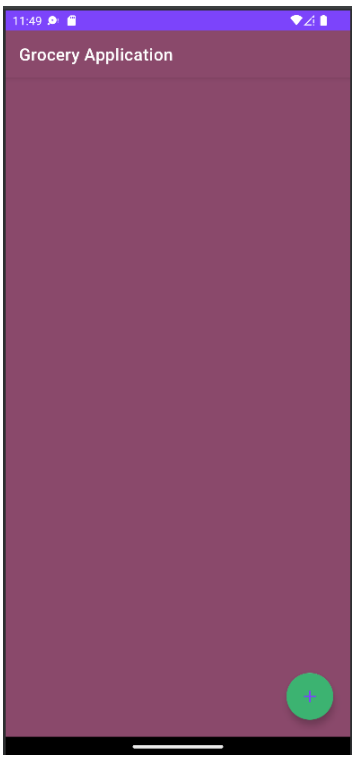
Note :- Since the page limit is exceeding I can't put the whole source code here . Please check the drive link or the github link below for full code.

Github link: https://github.com/Aksh08

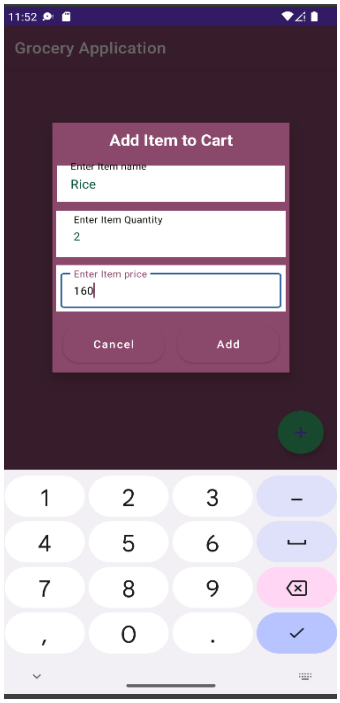
RESULT:

a) Logo of Grocery application.

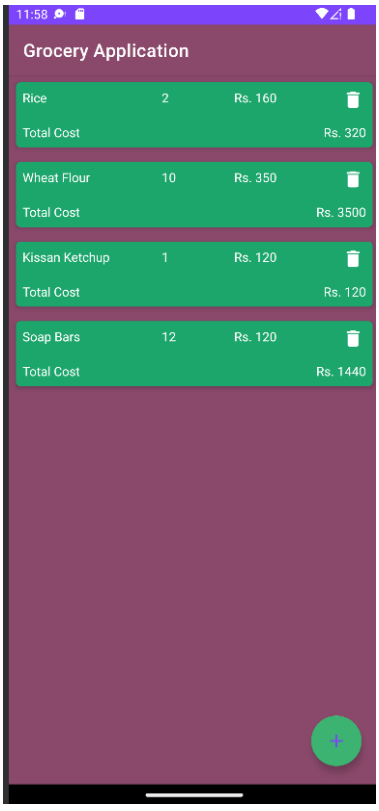
b) Layout of Home screen of Grocery Application.



c) Adding items to Cart.



d) Cart after adding Items in it.



ADVANTAGES:

- User can purchase grocery products through his mobile phones that support android.
- User does not have to wait in long queue and does not have to struggle with trolleys.
- User can coolly sit at home and purchase the products according to his like.

DISADVANTAGES:

- This system won't work in mobile phone that does not support android.
- Product quality would differ from manually purchasing product in a supermarket.

APPLICATIONS:

- This application can be used by any user who loves to shop and this application can be used by many house wives.

CONCLUSION: As people are shifting to online ordering, it is a good idea to develop an online grocery delivery app development. You need to take a few steps to easily overcome the challenges that are being faced by the remaining owners of the grocery business.

This project will be helpful to larger masses of people. The project is user friendly and can make improvements based on the user requirements. The project will be more useful in today's busy world. The project is made in a realistic method with proper security enhancements.

Reference:

● Google: google

● Geeksforgeeks: gfg

● Android Developer: android_devop

● YouTube: https://www.youtube.com/watch?v=OpQ3VzzgE0g&t=7s

●SmartInternz: https://smartinternz.com/

Name: Akshat Singhal
SBID: SB20220204388