GROCERY LIST APPLICATION USING KOTLIN

IN ANDROID STUDIO

AN ANDROID PROJECT REPORT

SUBMITTED BY:

JAYASHREE G

UNDER

SPSAPL20220112577

Virtual Internship Android Application Development Using Kotlin

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A smartphone can be used astonishingly well for grocery store lists. You always have it with you. You always have your phone with you. It is sensible to include your grocery list there rather than lugging around a second item. Fortunately, there are a tonne of options available.

Using Android Studio, we'll create an application for groceries for Android. Because we can't remember everything, we frequently forget to buy the things we want to buy. However, with the aid of this app, you may make a list of the groceries you intend to buy so that you don't forget anything. Recycler View, Coroutines, and (MVVM) architectural patterns are used in this project to display the list of objects. Room is the database. It effectively handles the essentials. When usual, you add things to the list and then take them off as you make purchases. But this one also features full offline support.

1.1 Abstract

Many people find grocery shopping to be boring and uninteresting, but it is an important activity to carry out because it is essential to human lifestyle. Simple methods are employed to help carry out these chores, some of which include writing down the things that need to be bought on paper or a mobile device or, most frequently, making a mental list. The best way to solve the drawbacks of paper-based grocery lists is to construct digital grocery lists utilising mobile apps, as smartphones perform a variety of functions and becoming an increasingly important part of people's life.

This study introduces "Grocery List," a mobile software solution that helps consumers to easily overcome the aforementioned difficulties when performing their grocery shopping. The application is made up of a number of modules, including an interactive shopping list that lets users add and remove products. The Smart Shopping List is a mobile-assisted software application that allows consumers to elevate their grocery shopping experience to a new level. This current work highlights our efforts on this project. Users are able to make shopping lists thanks to it.

1.2 Purpose

The primary goal of this project is to list the products so that users won't be able to forget them when they go to the grocery shop. This grocery list application helps users manage the daily chaos of remembering the grocery list more easily.

Making an app that stores the user's grocery list items in a list and allows for the addition and deletion of newly added items is the aim of this project. I want to create a trustworthy system, and I have the following objectives:

- Create a system that allows users to enter product information like name, quantity, and price.
- Create a database room to hold user data that has already been added to a list by the user. This room will also allow the user to delete previously added items from the list.
- Create a user interface that is user friendly. Create a solid user interface that is compatible with all Android devices.

CHAPTER 2

LITERATURE SURVEY

2.2 Existing Problem

Despite the fact that many people view grocery shopping as a boring and uninteresting task, it is an essential activity and a required component of daily living. Grocery buying is regarded as a sophisticated daily living skill that requires a person to acquire and practise a few processes, including planning, carrying out the shopping, and evaluating the actions that follow. In order to promote independent living, it should be taught to individuals with disabilities because it is a challenging skill to learn. Despite the fact that many individuals shop as a leisure activity, many of them do not like the food shopping process. Grocery shopping is a routine that can happen every day, every week, or every month.

According to the previous study, a grocery shopping list includes three stages: the development stage, the fulfilment (or usage) stage, and the outcome (or post-purchase) stage. The success of the grocery shopping process is determined by the development stage, which entails the process of creating a grocery list that contains a list of products that individuals need to buy. Most of the time, individuals go grocery shopping with a list in hand to assist them remember what they need to buy for their home. Making a grocery list involves doing pre-shopping tasks including checking the cupboard for food and planning the week's worth of meals to determine what needs to be purchased. Given how boring and time-consuming these tasks are, as well as how busy people's lives are, it can be challenging to create a shopping list. Even though creating a grocery list necessitates several additional tasks, they are typically underestimated since they go unmet and unacknowledged. Previous research has shown that grocery lists have an impact on many aspects of daily living. Grocery lists might encourage consumers to adopt a better diet and experience reduced body mass index. In goal-directed shopping, people make grocery lists for the things they need to keep the budget in check and prevent frivolous spending. Additionally useful for determining consumers' prepurchase intentions are studies on grocery lists.

A shopping list may include goods for cleaning, caring for pets, buying meat, dairy products, fresh produce, and personal care items. Other family members besides the person who made the list, such as the daughters and sons, may eat the things (usually the mother). When creating a shopping list, the creator must go over the stock of groceries and the level of utilisation at the time. Because of how time-consuming this procedure is, the shopping items might be overlooked and left off the list.

A grocery list is regarded as a supportive tool for the process of food shopping. It may be made by either recalling the items in mind or writing them down on a sheet of paper. Making a shopping list is frequently done by writing down the things that need to be purchased on a piece of paper. The approach has been around for a while, but a paper grocery list may be quickly ruined by liquids and dirt, is hard to see because of handwriting, and can even be left at home or lost.

2.2 Proposed solution

Shopping with a list helps you remember things and helps you stay away from impulsive buys. Additionally, it serves as a planning tool for managing household spending on groceries and food as well as meal planning. The advancement of smartphones and mobile technology has provided a way to get around the drawbacks of paper shopping lists by enabling the creation and management of grocery lists using mobile apps in a more convenient and adaptable manner. Given that smartphones now perform a variety of functions and are progressively becoming a necessary element of daily life, it is the most current answer. Consumer behaviour studies have placed a lot of emphasis on mobile-assisted grocery shopping, which has led to the creation of hybrid, intelligent, and multimodal digital shopping lists that are meant to aid customers with their shopping activities. Instantaneously prepared and saved in each user's smartphone, the digital grocery lists may be quickly accessed for changing the products or using as a guide during grocery shopping.

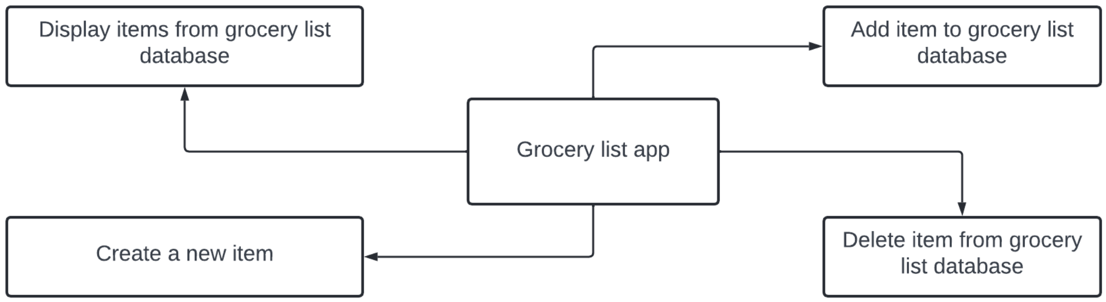
CHAPTER 3

THEORETICAL ANALYSIS

The project for the grocery list application will assist the user or admin in storing the list of products in the correct order. The list's entries can be added and removed at the user or administrator's discretion.

- UI design in the android platform
- Android application development
- Database connection to store user data

3.1 Block diagram


Display items from grocery list database
Add item to grocery list database
Grocery list app
Create a new item
Delete item from grocery list database

3.2 Hardware requirements

- 64-bit Microsoft® Windows® 8/10/11.
- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
- 8 GB RAM or more.
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolution.

3.3 Software requirements

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

- Operating System: Windows 11
- language: Kotlin
- Emulator: Pixel 4 API 30

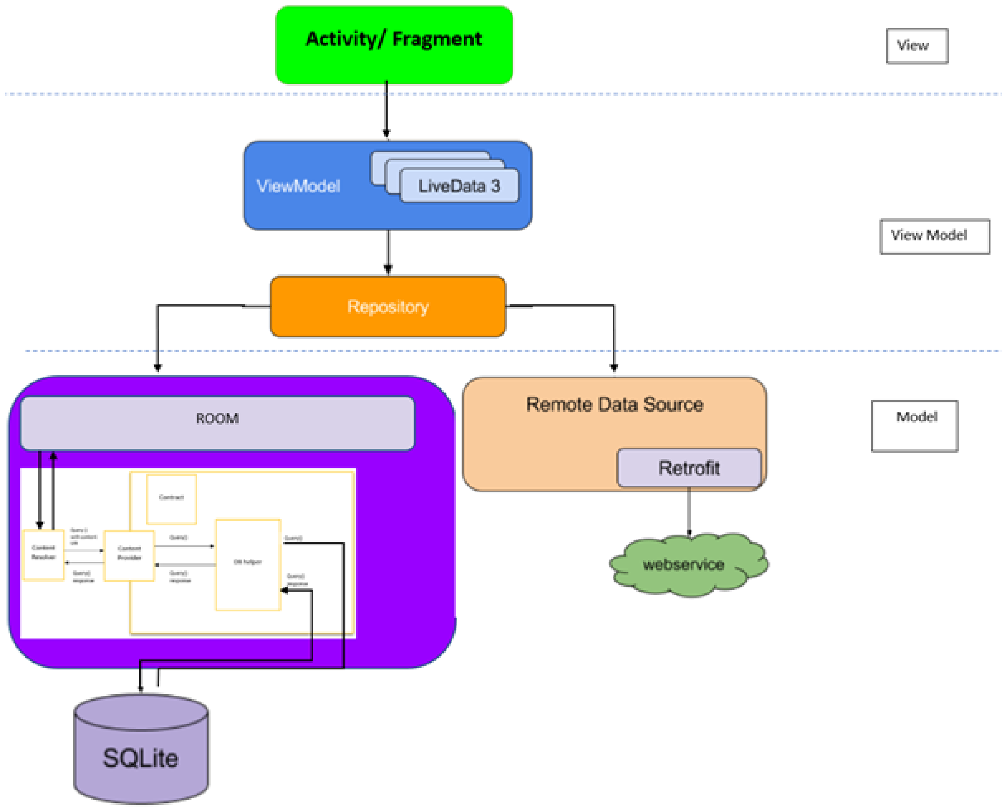
CHAPTER 4

IMPLEMENTATION AND DESIGN

Coroutines, Model View ViewModel (MVVM) architectural patterns, Room for the database, and RecyclerView to show the list of objects are all used in this project.

4.1 MVVM

Android uses MVVM architecture to arrange project code and make it simpler to comprehend. An architectural design pattern used in Android is MVVM. XML files and Activity classes are treated as Views by MVVM. With this design approach, UI and logic are entirely separated. Here is a visual representation of MVVM.



Flow diagram

4.2 Room database

The data of apps, such as the name, amount, and price of groceries, are stored in the Room persistence library, a database management library. Room is a cover layer for SQLite that makes it easier to operate on databases.

<u>4.3 RecyclerView</u>

RecyclerView is a container that is used to show a collection of data in a sizable data set that can be scrolled efficiently by keeping the number of views to a minimum.

<u>4.4 Coroutines</u>

We utilise coroutines, a lightweight thread, to operate on other threads so that our software doesn't crash or cause our main thread to halt.

<u>4.5 Algorithm</u>

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 Primary Key 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 GrocerylistEntities. 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. Right-click on com.example.application-name go to new and create Kotlin file/class and name the file as GrocerylistDao. 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 GrocerylistRepository. 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.
13
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 GrocerylistViewModel (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 GrocerylistViewModelFactory.

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 GrocerylistAdapter. We will code adapter class for recycler view. In the GrocerylistAdapter 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.application-name. 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 GrocerylistAdapter.

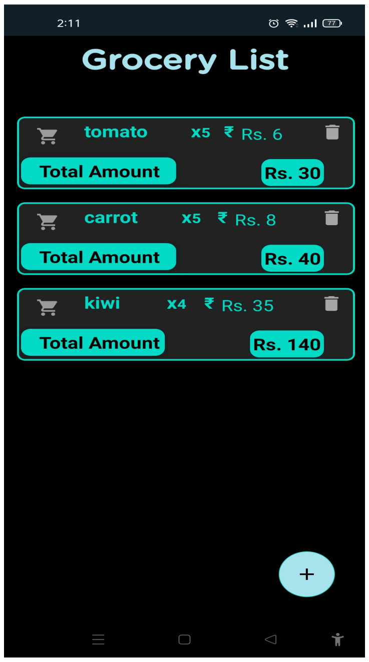
Step 7: To enter grocery item, quantity, and price from the user we have to create an interface. To implement this interface we will use DialogBox. First create UI of dialog box. In this dialog box we will add three edit text and two text view. Three edit text to enter grocery item name, quantity and 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.application-name > 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.

CHAPTER 5

CONCLUSION

5.1 Android implementation result



2:11

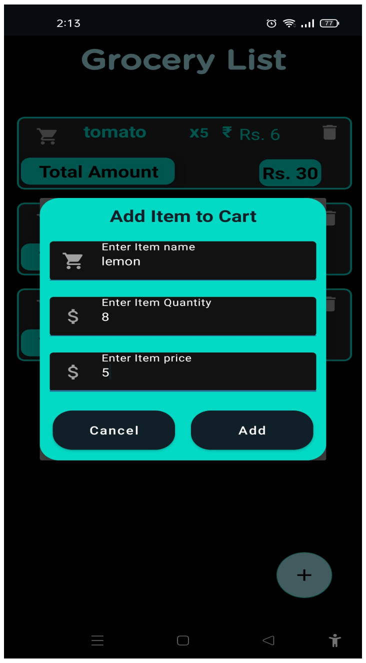
Grocery List

tomato x5 ₹ Rs. 6
Total Amount Rs. 30

carrot x5 ₹ Rs. 8
Total Amount Rs. 40

kiwi x4 ₹ Rs. 35
Total Amount Rs. 140

+



2:13

Grocery List

tomato x5 ₹ Rs. 6
Total Amount Rs. 30

Add Item to Cart

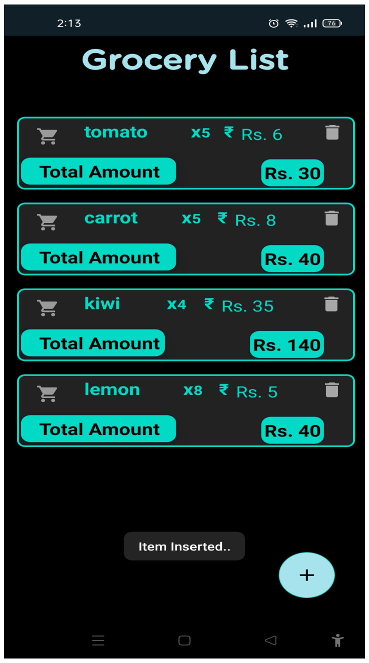
Enter Item name: lemon

Enter Item Quantity: 8

Enter Item price: 5

Cancel | Add

+



2:13

Grocery List

tomato x5 ₹ Rs. 6
Total Amount Rs. 30

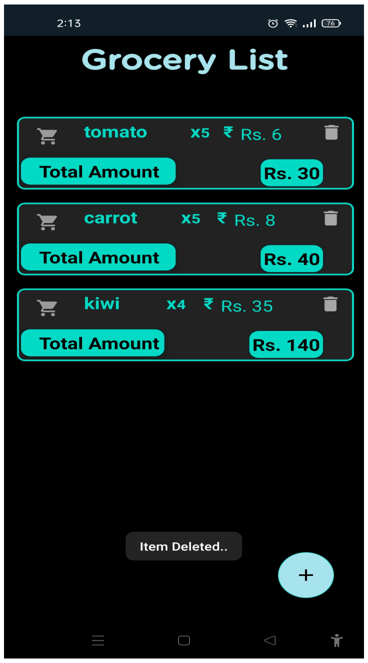
carrot x5 ₹ Rs. 8
Total Amount Rs. 40

kiwi x4 ₹ Rs. 35
Total Amount Rs. 140

lemon x8 ₹ Rs. 5
Total Amount Rs. 40

Item Inserted..

+



2:13

Grocery List

tomato x5 ₹ Rs. 6
Total Amount Rs. 30

carrot x5 ₹ Rs. 8
Total Amount Rs. 40

kiwi x4 ₹ Rs. 35
Total Amount Rs. 140

Item Deleted..

+

5.2 Conclusion

Aspects of performance and security were taken into account from the solution's conception and at every level of development. The size of the application is around 7MB. As a result, installing the application on a device doesn't demand a lot of storage space. Additionally, this application supports offline use. To make sure that the software programme is created in a way that the usability of the application has been properly maintained, a usability analysis for the Grocery List has also been carried out.

The mobile application allowed many users to collaborate on the creation and management of grocery lists. The analysis of the mobile app revealed intriguing results that might inspire additional research into enhancing the design and communication processes of mobile apps to assist the administration of grocery lists among family members.

5.3 Future work

Based on the current state of the grocery list app, a potential future implementation would be to broaden the app's scope entirely. For instance, I may expand the app's aim to categorise grocery goods rather than just having it solely for grocery lists. In this manner, the project might have more applications and conceivably appeal to more people.

CHAPTER 6

BIBLIOGRAPHY

6.1 References

- https://ideaexchange.uakron.edu/cgi/viewcontent.cgi?article=1690&context=honors_research_projects
- https://www.youtube.com/watch?v=vdcLb_Y71Ic
- https://www.geeksforgeeks.org/how-to-build-a-grocery-android-app-using-mvvm-and-room-database/
- https://developer.android.com/courses/android-basics-kotlin/course

6.2 Acknowledgements

I put a lot of work into this project. However, without the kind support and assistance of many people and organisations, it would not have been feasible. I want to express my sincere gratitude to each and every one of them.

I am very grateful to SMARTINTERNZ (Experiential Learning & Remote Externship Platform to bring academia & industry close for a common goal of talent creation) for their direction and constant supervision as well as for providing important information regarding the project and also for their support in completing the project. I want to thank the members of (Smart Internz) for their kind encouragement and cooperation, which enabled me to finish this project. I want to offer my sincere appreciation and thanks to those in the profession who took the time to pay me such close attention.

APPENDIX

A. Source code

Github URL:

https://github.com/smartinternz02/SPSGP-103095-Virtual-Internship---Android-Application-Development-Using-Kotlin

Smart Internz Registered ID:

jayashreeekk@gmail.com