

Android Development Project Report

On Grocery List Application Using Kotlin and GCP



Done By,
Vijay Chandar

Introduction to Android Development:

Android operating system is the largest installed base among various mobile platforms across the globe. Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world. It conquered around 71% of the global market share by the end of 2021, and this trend is growing bigger every other day. The company named Open Handset Alliance developed Android for the first time that is based on the modified version of the Linux kernel and other open-source software. Google sponsored the project at initial stages and in the year 2005, it acquired the whole company. In September 2008, the first Android-powered device was launched in the market. Android dominates the mobile OS industry because of the long list of features it provides. It's user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android-compatible smartphones. As a result, the market observes a sharp increase in the demand for developing Android mobile applications, and with that companies need smart developers with the right skill set. At first, the purpose of Android was thought of as a mobile operating system. However, with the advancement of code libraries and its popularity among developers of the divergent domain, Android becomes an absolute set of software for all devices like tablets, wearables, set-top boxes, smart TVs, notebooks, etc.

Features of Android

- Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.
- Android Open Source Project so we can customize the OS based on our requirements.
- Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
- Using Wi-Fi technology we can pair with other devices while playing games or using other applications.
- It contains multiple APIs to support location-tracking services such as GPS. We can manage all data storage-related activities by using the file manager.
- It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video. It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
- It supports multimedia hardware control to perform playback or recording using a camera and microphone.
- Android has an integrated open-source WebKit layout-based web browser to support User Interfaces like HTML5, and CSS3.
- Android supports multi-tasking means we can run multiple applications at a time and can switch between them. It provides support for virtual reality or 2D/3D Graphics.

Android Versions The following table shows the version details of android which is released by Google from 2007 to date.

Code Name	Version	API level	Release date
–	Android 1.0	1	September 23, 2008
–	Android 1.1	2	February 9, 2009
Cupcake	Android 1.5	3	April 30, 2009
Donut	Android 1.6	4	September 15, 2009
Eclair	Android 2.0 – 2.1	5–7	October 26, 2009
Froyo	Android 2.2 – 2.2.3	8	May 20, 2010
Gingerbread	Android 2.3 – 2.3.4	9–10	December 6, 2010
Honeycomb	Android 3.0.x – 3.2.x	11 – 13	February 22, 2011
Ice Cream Sandwich	Android 4.0 – 4.0.4	14 – 15	October 18, 2011
Jelly Bean	Android 4.1 – 4.1.2	16 – 18	July 9, 2012
Kitkat	Android 4.4 – 4.4.4	19	July 9, 2012
Lollipop	Android 5.0 – 5.1	21 – 22	October 17, 2014
Marshmallow	Android 6.0 – 6.0.1	23	October 5, 2015
Nougat	Android 7.0 – 7.1	24 – 25	August 22, 2016
Oreo	Android 8.0	26	August 21, 2017
Pie	Android 9.0	27	August 6, 2018
Android Q	Android 10.0	29	September 3, 2019
Android 11	Android 11.0	30	September 8, 2020
Snow Cone	Android 12.0 – 12.1	31–32	October 4, 2021
Tiramisu	Android 13		UPCOMING

Programming Languages used in Developing Android Applications

- Java
- Kotlin

Developing the Android Application using Kotlin is preferred by Google, as Kotlin is made an official language for Android Development, which is developed and maintained by JetBrains. Previously before Java is considered the official language for Android Development. Kotlin is made official for Android Development in Google I/O 2017.

Advantages of Android Development

- The Android is an open-source Operating system and hence possesses a vast community for support.
- The design of the Android Application has guidelines from Google, which becomes easier for developers to produce more intuitive user applications.
- Fragmentation gives more power to Android Applications.
- This means the application can run two activities on a single screen.
- Releasing the Android application in the Google play store is easier when it is compared to other platforms.

Disadvantages of Android Development

- Fragmentation provides a very intuitive approach to user experience but it has some drawbacks, where the development team needs time to adjust to the various screen sizes of mobile smartphones that are now available in the market and invoke the particular features in the application.
- The Android devices might vary broadly. So the testing of the application becomes more difficult.
- As the development and testing consume more time, the cost of the application may increase, depending on the application's complexity and features.

Kotlin Android Basics

Kotlin is a statically typed, general-purpose programming language developed by JetBrains that has built world-class IDEs like IntelliJ IDEA, PhpStorm, Appcode, etc. It was first introduced by JetBrains in 2011 and a new language for the JVM. Kotlin is object-oriented language, and a “better language” than Java, but still be fully interoperable with Java code. Kotlin is sponsored by Google, announced as one of the official languages for Android Development in 2017.

Kotlin Hello World Program

```
// Kotlin Hello World Program
fun main(args: Array<String>) {
    println("Hello, World!")
}
```

Key Features of Kotlin

Statically typed – Statically typed is a programming language characteristic that means the type of every variable and expression is known at compile time. Although it is statically typed language, it does not require you to explicitly specify the type of every variable you declare.

Data Classes– In Kotlin, there are Data Classes which lead to auto-generation of boilerplate like equals, hashCode, toString, getters/setters and much more.

Concise – It drastically reduces the extra code written in other object-oriented programming languages.

Safe – It provides the safety from most annoying and irritating NullPointerExceptions by supporting nullability as part of its system. Every variable in Kotlin is non-null by default. *Interoperable with Java* –

Kotlin runs on Java Virtual Machine (JVM) so it is totally interoperable with java. We can easily access use java code from kotlin and kotlin code from java.

Functional and Object Oriented Capabilities – Kotlin has rich set of many useful methods which includes higher-order functions, lambda expressions, operator overloading, lazy evaluation, operator overloading and much more. Higher order function is a function which accepts function as a parameter or returns a function or can do both.

Smart Cast – It explicitly typecasts the immutable values and inserts the value in its safe cast automatically.

Compilation time – It has higher performance and fast compilation time.

Tool- Friendly – It has excellent tooling support. Any of the Java IDEs – IntelliJ IDEA, Eclipse and Android Studio can be used for Kotlin. We can also be run Kotlin program from command line.

Advantages of Kotlin Language

- Easy to learn - Basic is almost similar to java. If anybody worked in java then easily understand in no time.
- Kotlin is multi-platform – Kotlin is supported by all IDEs of java so you can write your program and execute them on any machine which supports JVM.
- It's much safer than Java.
- It allows using the Java frameworks and libraries in your new Kotlin projects by using advanced frameworks without any need to change the whole project in Java.
- Kotlin programming language, including the compiler, libraries and all the tooling is completely free and open source and available on GitHub.

Applications of Kotlin Language

- You can use Kotlin to build Android Application.
- Kotlin can also compile to JavaScript, and making it available for the frontend.
- It is also designed to work well for web development and server-side development.
- Native App Development
- Desktop App Development
- WebDevelopment
- Cross-platform mobile app development

Index

1 INTRODUCTION

1.1 Overview

A brief description about your project

1.2 Purpose

The use of this project. What can be achieved using this.

2 LITERATURE SURVEY

2.1 Existing problem

Existing approaches or method to solve this problem

2.2 Proposed solution

What is the method or solution suggested by you?

3 THEORITICAL ANALYSIS

3.1 Block diagram

Diagrammatic overview of the project.

3.2 Hardware / Software designing

Hardware and software requirements of the project

4 EXPERIMENTAL INVESTIGATIONS

Analysis or the investigation made while working on the solution.

5 FLOWCHART

Diagram showing the control flow of the solution

6 RESULT

Final findings (Output) of the project along with screenshots.

7 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

8 APPLICATIONS

The areas where this solution can be applied

9 CONCLUSION

Conclusion summarizing the entire work and findings.

10 FUTURE SCOPE

Enhancements that can be made in the future.

11 BIBLIOGRAPHY

References of previous works or websites visited/books referred for analysis about the project, solution previous findings etc.

CHAPTER 1: Introduction

1.1 ABSTRACT

Some people include shopping as one of their daily activities, while others don't even give it a second thought. This comparison reveals the issues people have when shopping. Limited time, foreign residents without cars, a transportation problems, the belief that physical shopping is a waste of time, health concerns, and a distance from the market are just a few of the issues people face when shopping. additionally, some items are hard to come by. We have researched our concept, which is related to personal shopping, as well as the issues mentioned above. As a result, we created a program that combines various market stores, i.e. (Malls, supermarkets, and pharmacies). This cutting-edge app for personal grocery shopping enables customers to get everything they need and makes recommendations based on past purchases.

Then deliver items to their doorstep and facilitate online shopping so that customers can simultaneously browse an unlimited number of products. Instead of physically wasting their time, this work encourages people to make the most of it by being safer and more approachable. Additionally, rather than traveling great distances to shop, people can order the item online from the comfort of their own homes. Additionally, this app might aid those who are experiencing health issues and are unable to physically purchase items in order to prevent further issues. Last but not least, some people must keep up with technological advancements because they lack access to transportation for shopping.

1.2 OBJECTIVE

The main goal of this project is to list the items so that users won't forget them when they go to the grocery store. This grocery application also makes it easier for users to manage their daily chaos.

1.3 PROBLEM TARGETED

We decided to build an app to store the items in the database for their future use because it's difficult for the users to remember every item in this busy lifestyle and they frequently can't recall their required necessities. Users can remove the previously added items from the database after purchasing the items.

1.4 PROBLEM'S PRIMARY GOALS

This project aims to create an app that stores user items in a cart and allows users to edit and remove newly added items from the list. 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 where users can store information that has already been added to their shopping carts and from which they can also remove items.
- Create a user-friendly UI design that is effective.
- Create a solid user interface that works with every Android device.

1.5 INTRODUCTION

Using Android Studio, we'll create an application for groceries on 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 can make a list of the groceries you intend to buy so that you don't forget anything. Room for database, RecyclerView, Coroutines, and (MVVM) architectural patterns are used in this project to display

the list of items.

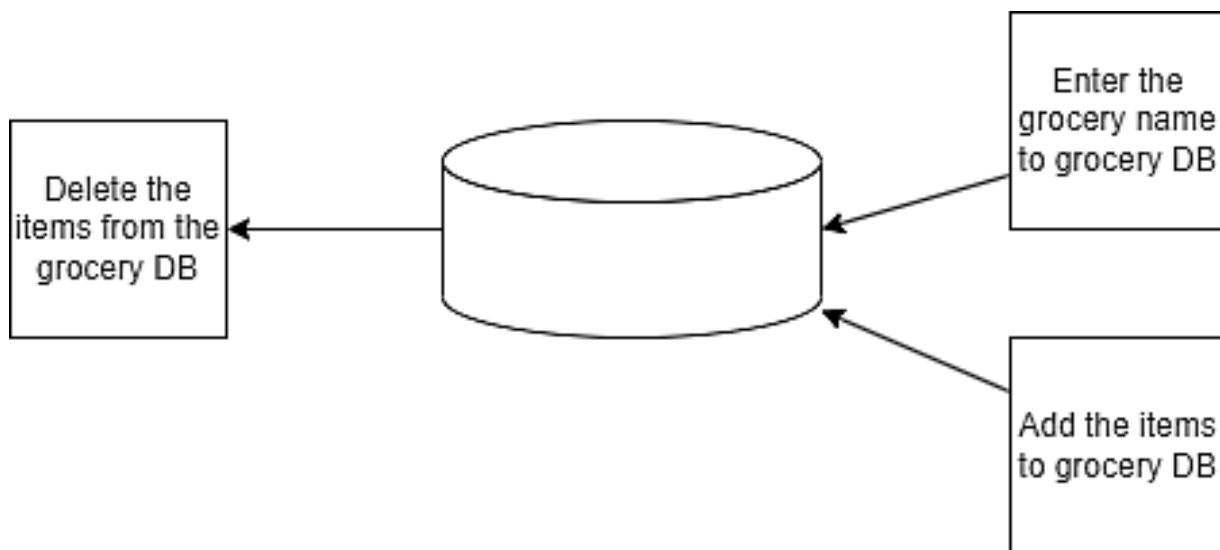
CHAPTER 2: Background & Diagrams

2.1 BACKGROUND

The project for the grocery cart application will assist the user or administrator in organizing the list of items into the correct order. The list's items can be added and removed at the user's or administrator's discretion.

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

2.2 CONTEXT DIAGRAM



CHAPTER 3: Technical Requirements

3.1 SOFTWARE

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

- Operating System: Windows 11
- Software: Kotlin and Java
- Emulator: Pixel 4 API 30

3.2 HARDWARE

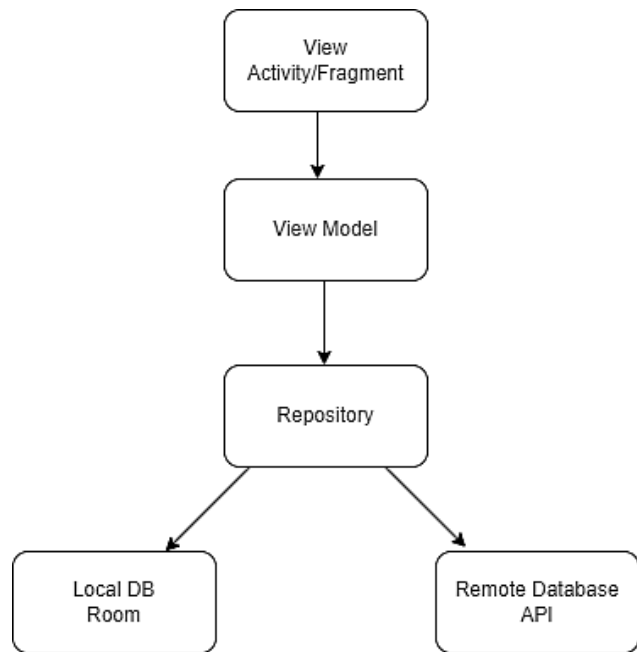
- RAM: 16 GB RAM
- Enter the item names to Grocery DB
- ROM: 20 GB ROM

CHAPTER 4: Implementation and Designing

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

MVVM (Model View ViewModel)

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



ROOM Database

The data of apps, such as the name, quantity, 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.

RecyclerView

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.

Coroutines

We use coroutines, a lightweight thread, to operate on other threads so that our app doesn't crash or cause our main thread to block.

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 you 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 GroceryEntities. See the code below to understand and implement it completely.

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 is 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 them with @Insert, the second is for deleting items from the database annotated with @Delete and the 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 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 associated with the database, and the 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 is used as an interface between View and Data. Grocery View Model class inherits 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 the adapter class for the 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 the Adapter package and create a Kotlin file/class name it GroceryAdapter.

Step 7:

To enter grocery items, quantities, and prices from the user we have to create an interface. To implement this interface we will use DialogBox. First, create the UI of the dialog box. In this dialog box, we will add three edit text and two text views. Three edit text to enter the grocery item name, quantity, and price. Two text views one for save and the other for cancel. After clicking the save text all data is saved into the database and 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

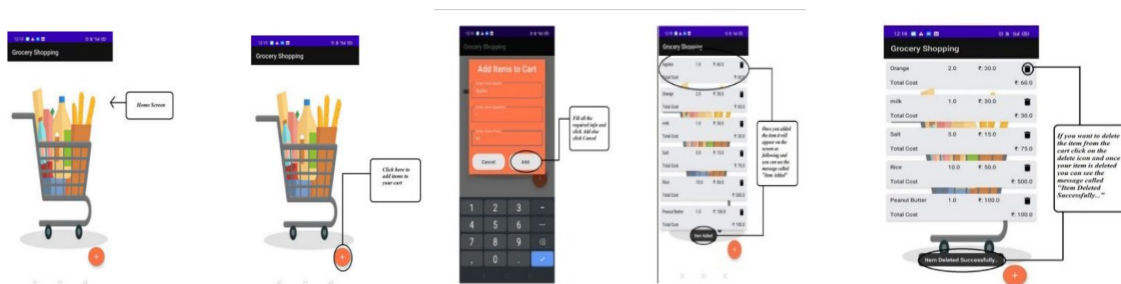
clicklistener on save text we have to create an interface first in which we create a function. Go to the app > java > com.example.applicationname > UI. Right-click on the UI package and create a Kotlin file/class and create an interface named as DialogListener.

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 a click listener on add button to open the dialog box.

CHAPTER 5: Conclusion and Future Scope

5.1 Expected Outcome



5.2 Conclusion & Future Scope

This grocery application will help to store the list of data items including the name of the item, price and quantity required. Admins store his/her data in the list, and the grocery application is very helpful to users.

Future Scope:

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

The Features are:

- Add User Panel
- Add Admin Panel
- Provide Login Authentication
- Add Image to user Product and Rating

CHAPTER 6: URLs, Ids, Acknowledgements, Reflection Notes and References

6.1 URLs & Account Ids

GitHub URL: <https://github.com/smartinternz02/SPSGP-54858-Virtual-Internship---AndroidApplication-Development-Using-Kotlin>

Smart Internz Registered ID: vijaychandar3002@gmail.com

Demo Link: <https://www.youtube.com/embed/etKDbtO233Y>

6.2 Acknowledgements

I have taken many 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 to members of (Smart Internz) for their kind co-operation and encouragement which help me in the completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time. My thanks and appreciations also go to people who have willingly helped me out with their abilities

6.3 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. I have learned to work well as a team and that without my counter parts the work would not get done. Another aspect that I learned throughout the internship is to never be afraid to ask lots of questions. By asking questions we get answers.

6.4 References

<https://github.com/divyanshu15/GroceryApp>
<https://youtu.be/5YmJLB8f3W0>
<https://github.com/smartinternz02/SPSGP-68275-Virtual-Internship---Android-Application-Development-Using-Kotlin>
https://www.youtube.com/watch?v=vdcLb_Y71lc
<https://www.geeksforgeeks.org/introduction-to-androiddevelopment/#:~:text=Google%20first%20publicly%20announced%20Android,with%20the%20version%20Android%201.0>
<https://www.geeksforgeeks.org/introduction-to-kotlin/>