ABSTRACT

I went through a practical Industrial training in SmartInternz, on **Android Application Development Using Kotlin** which included the study of Application architecture, Cloud Platform and common UI interactions. It was a great opportunity as well as a nice experience to get to know about the real time working of the Android Applications using Kotlin.

I had a great learning experience where I got to look and watch the real time working of the Android Applications. Each aspect of the visit was very fascinating and intriguing as I was allowed to interact with the every component of the entire Application. It was very good and very different real time experience for me where we faced practical issues on APP Development.

As demand for skilled Android developers increases in the job marketplace, there is an greater need for the next generation to develop the skills of Android Development.

This program will give to complete hands-on experience on android application development using Kotlin also you will develop an application on your own.
The Project is a Grocery List Application.

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. The goal of this project is making an app which store the user items in cart and user can modify and delete the added item in list.

INDEX

CONTENTS **page**

CHAPTER-I

INTRODUCTION

SmartBridge and Google have teamed up to create an outcome-driven skilling effort that will train 2000+ educators and 5000 students on android application development in Kotlin programming. This program has been recognised by AICTE for delivery as a virtual internship program to all higher education students in India. This Program will be executed in two phases, phase-1 is to teach the educators on android skills and application lifecycle management and qualify them as "Mentor on Campus" to drive the phase-2 of the program in campus called virtual internship program. The virtual internship program is 100+ Hrs. experiential learning program containing hands-on bootcamps, courses, learning resources and project work.

1.1 Overview:

Android Studio is an open and free which is used to develop the android application for Mobile's, TV's, Watches and other Android OS. It provides leverage the power of SQL Database and build a app that shows items added by user. The Project is a Grocery List Application. 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.

1.2 Purpose:

The goal of the my project is making an app which store the user items in cart and user can modify and delete the added item in list. To develop a reliable system , I have some specific goals such as:

1. Develop a system such that user can add item details like product name, product Quantity and Product Price.
2. Develop a database room which is used to store the user data which already added by the user in cart and user can also remove the previous added item in cart.
3. Develop the good UI design which user friendly to user.
4. Develop the good UI which is supported for all android devices.

CHAPTER-2

LITERATURE SURVEY

This section explores the technologies which were used in this project. The section first discusses the key concepts for Android Applications, followed by the metrics used to evaluate them along with the environments and the libraries used to complete this project.

2.1 Existing System:

There are some grocery apps that are used for shopping grocery items.It also locate the nearby shops to buy Groceries. But it does not list the quantity and price of the grocery items that we want to buy and it does not help us to us to remember things.

2.2 Proposed System:

Grocery app is a project that will help the user or admin to store the list of items in sequence order. If user want to add extra items, it is possible and if user want to remove the previous items, it is also available.

- UI DESIGN IN ANDROID PLATFORM
- ANDROID APPLICATION DEVELOPMENT
- DATABASE CONNECTION TO STORE USER DATA

2.3 Studies:

1. Design user friendly environment
2. Connecting database
3. Add material icon for good UI
4. Create the vectors and Material Colours
5. Navigation of user action from one page to another page

2.4 Challenges:

Working with database friendly UI design for UX Gradle and settings

2.5 REQURIMENTS

2.5.1 FUNCTIONAL REQUIREMENTS:-

- Google Cloud Platform.
- API Key
- APK files
- Emulator

2.5.2 NON FUNCTIONAL REQUIREMENTS

- Reliability.
- Performance
- Usability.
- Platform independent
- Supportability

SOFTWARE AND HARDWARE REQUIREMENTS:

Software: The Software Package is Developed Using Kotlin and Android Studio. Basic SQL Commands Used to Store the Database.

Operating System: Windows 11
Software: Kotlin and Java
Emulator: Pixel 5 (Api 30) 3.2

Hardware:

RAM : 8GB RAM
ROM : 20GB

CHAPTER-3

EXPERIMENTAL ANALYSIS

Experimental Investigations 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.

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.

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

RecyclerView 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 coroutines to perform an operation on other threads, by this our main thread doesn't block and our app doesn't crash.

CHAPTER-4

PROBLEM IDENTIFICATION

Build my first Android apps with the Kotlin programming language. Add images and text to your Android apps, and learn how to use classes, objects, and conditionals to create an interactive app for our users.

Layouts

Improve the user interface of your app by learning about layouts, Material Design guidelines, and best practices for UI development.

Navigation

Enhance our users' ability to navigate across, into and back out from the various screens within your app for a consistent and predictable user experience.

Connect To The Internet

Write coroutines for complex code, and learn about HTTP and REST to get data from the internet.

Data Persistence

Keep our apps working through any disruptions to essential networks or processes for a smooth and consistent user experience.

Work Manager

Use Android Jetpack's Work Manager API to schedule necessary background work, like backing up data or downloading fresh content, that keeps running even if the app exits or the device restarts.

Create A Project In Cloud Console

In this milestone you will be working with the API. You need to create an project & enable the place API

Build The App

In this milestone you need get started with the application building part.

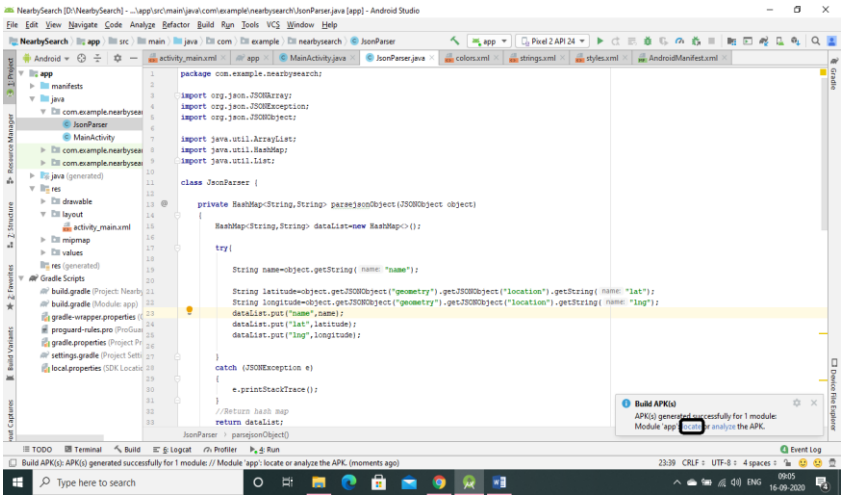
Build The APK File

In this activity, I will be starting building the apk file my application.

Steps to be followed to build the application

Click on Build Build--->Build Bundle/APK(s)--->Build APK.

Gradle will build the APK file. Click on locate which will open the APK file path.



The output APK file is obtained, and you share the file to android mobile and install it.

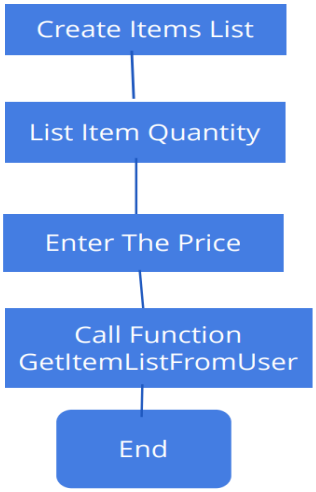
You can test apps with an emulator or you can connect your Android phone to the laptop and build the app.

CHAPTER-5

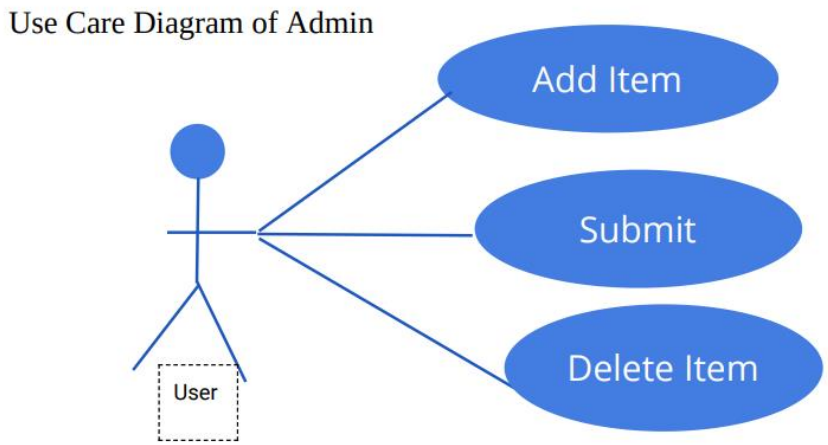
SYSTEM DESIGN

This chapter provides information of software development life cycle, design model i.e., various UML diagrams and process specification.

FLOWCHART:



USECASE DIAGRAM:



CHAPTER-6

IMPLEMENTATION

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

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.applicationname.
Right-click on com.example.application-name go to new and create Kotlin file/class and name the file as GroceryEntities.
See the code 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. Rightclick 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 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 tothe 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 tothe app > java > com.example.applicationname.

Right-click on com.example.applicationname 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 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 GroceryViewModel (repository).
Go to the app > java > com.example.application-name > UI.
Right-click onthe 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 , 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 adapter class for 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 Adapter package and create a Kotlin file/class name it GroceryAdapter.

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 Grocery Dialog.

To add a click listener 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 name it as Dialog Listener. In this final step we will code in our Main Activity.

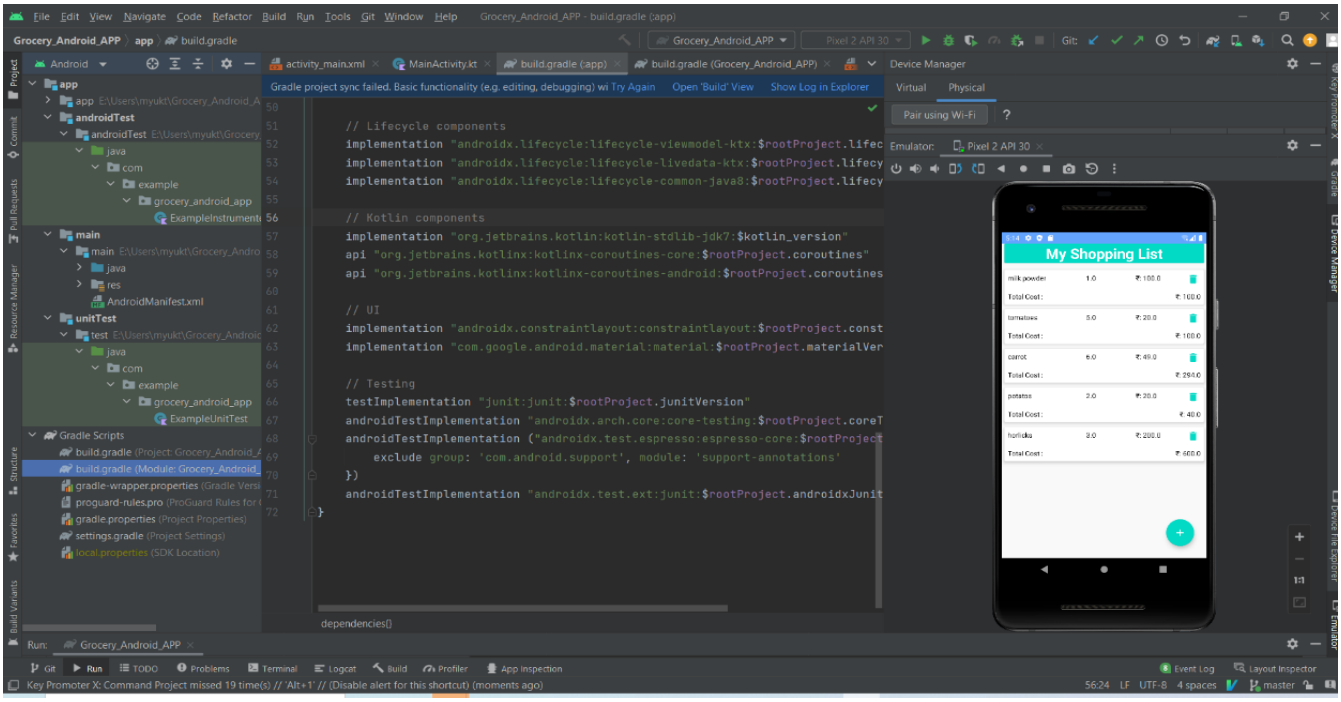
6.5 Sample Code

```
package
com.yukthapriya.grocerylist
                        import androidx.test.platform.app.InstrumentationRegistry
                        import androidx.test.ext.junit.runners.AndroidJUnit4
                        import org.junit.Test
                        import org.junit.runner.RunWith
                        import org.junit.Assert.*
                        /**
                        * Instrumented test, which will execute on an Android device.
                        *
                        * See [testing documentation](http://d.android.com/tools/testing).
                        */
                        @RunWith(AndroidJUnit4::class)
                        class ExampleInstrumentedTest {
                        @Test
                        fun useAppContext() {
                        // Context of the app under test.
                        val                          appContext                          =
                        InstrumentationRegistry.getInstrumentation().targetContext
                        assertEquals("com.androiddevs.grocerylist", appContext.packageName)
                        }
                        }
```

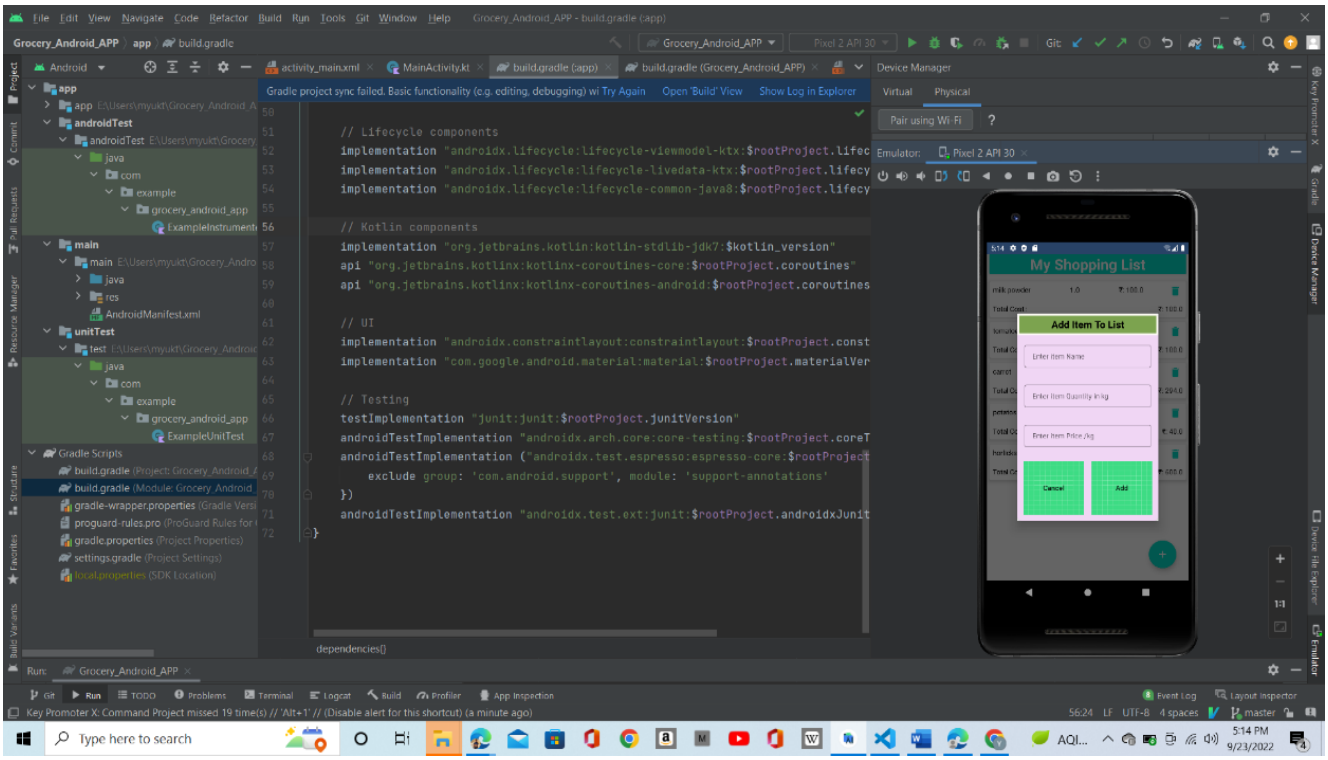
CHAPTER-7

RESULTS

4.1 HOME PAGE & LIST ITEMS:



4.2 POP UP PAGE



CHAPTER-8

CONCLUSION

This application will help us to store the list of items date, include name, price and quantity. Most of the shopping admins want to store his/her data in the list. This application is very helpfull for shop admins and also helpful for user to record the items and remember what items they bought.
This application Provides a “Quicklist” option, in which you can easily add items, create food groups, and create multiple shopping lists.

CHAPTER-9

FUTURE WORK

This application helps to store the list of items by Admin. In Future we can also add user panel which is add by user required item that are submitted to admin. The Features are:

1. Add User Panel
2. Add Admin Panel
3. Provide Login authentication
4. Add image to user product and rating.

CHAPTER -10

REFERENCES

Github Link: yukthapriya/Grocery_List-App (github.com)

Google Developer Profile Url: Google Developer Profile | Google Developers

https://developers.google.com/profile/u/yukthapriyamasupalli