**Simple Coffee App - Android Case Study**

Let's build a simple **Coffee Order App** in Android, which allows users to place an order for coffee, specifying the quantity, and choose options like cream and sugar. We'll focus on basic Android concepts such as **UI design with XML**, **Activity lifecycle**, **Intents**, and **SharedPreferences**.

**Features:**

1. Select the quantity of coffee.
2. Choose whether to add cream and/or sugar.
3. Calculate and display the total price.
4. Provide an order summary.
5. Optionally, save preferences using SharedPreferences.

**Tech Stack:**

* Language: **Kotlin**
* Android Version: **Android 5.0 (Lollipop)** or higher
* Tools: **Android Studio**

**1. App Architecture:**

* **MainActivity**: The main screen where the user selects coffee options.
* **SummaryActivity**: Displays the order summary.

**2. MainActivity Layout (XML)**

In this layout, we'll provide UI components for the user to select the quantity, cream, sugar options, and a button to place the order.

xml

Copy code

<!-- res/layout/activity\_main.xml -->

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/coffee\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Coffee Order"

android:textSize="24sp"

android:layout\_gravity="center"/>

<TextView

android:id="@+id/quantity\_label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Quantity:"

android:layout\_marginTop="16dp"/>

<EditText

android:id="@+id/quantity"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="number"/>

<CheckBox

android:id="@+id/cream\_checkbox"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add Cream"/>

<CheckBox

android:id="@+id/sugar\_checkbox"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add Sugar"/>

<Button

android:id="@+id/order\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Place Order"

android:layout\_marginTop="16dp"/>

</LinearLayout>

**3. MainActivity (Kotlin)**

In this activity, we will handle the button click event to calculate the total cost and pass the order summary to the next screen.

kotlin

Copy code

package com.example.coffeeapp

import android.content.Intent

import android.os.Bundle

import android.widget.Button

import android.widget.CheckBox

import android.widget.EditText

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val quantityEditText = findViewById<EditText>(R.id.quantity)

val creamCheckBox = findViewById<CheckBox>(R.id.cream\_checkbox)

val sugarCheckBox = findViewById<CheckBox>(R.id.sugar\_checkbox)

val orderButton = findViewById<Button>(R.id.order\_button)

orderButton.setOnClickListener {

val quantity = quantityEditText.text.toString().toIntOrNull()

if (quantity == null || quantity <= 0) {

Toast.makeText(this, "Please enter a valid quantity", Toast.LENGTH\_SHORT).show()

return@setOnClickListener

}

val hasCream = creamCheckBox.isChecked

val hasSugar = sugarCheckBox.isChecked

val pricePerCup = 5 // Basic coffee price

val totalPrice = calculateTotalPrice(quantity, hasCream, hasSugar, pricePerCup)

val summary = createOrderSummary(quantity, hasCream, hasSugar, totalPrice)

val intent = Intent(this, SummaryActivity::class.java).apply {

putExtra("ORDER\_SUMMARY", summary)

}

startActivity(intent)

}

}

private fun calculateTotalPrice(quantity: Int, hasCream: Boolean, hasSugar: Boolean, pricePerCup: Int): Int {

var totalPrice = quantity \* pricePerCup

if (hasCream) totalPrice += quantity \* 1 // Extra $1 for cream

if (hasSugar) totalPrice += quantity \* 1 // Extra $1 for sugar

return totalPrice

}

private fun createOrderSummary(quantity: Int, hasCream: Boolean, hasSugar: Boolean, totalPrice: Int): String {

return """

Order Summary:

Quantity: $quantity

Cream: ${if (hasCream) "Yes" else "No"}

Sugar: ${if (hasSugar) "Yes" else "No"}

Total Price: $${totalPrice}

Thank you for your order!

""".trimIndent()

}

}

**4. SummaryActivity Layout (XML)**

This layout will display the order summary passed from the MainActivity.

xml

Copy code

<!-- res/layout/activity\_summary.xml -->

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/order\_summary"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="18sp"/>

</LinearLayout>

**5. SummaryActivity (Kotlin)**

This activity receives the order summary and displays it on the screen.

kotlin

Copy code

package com.example.coffeeapp

import android.os.Bundle

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

class SummaryActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_summary)

val summaryTextView = findViewById<TextView>(R.id.order\_summary)

val orderSummary = intent.getStringExtra("ORDER\_SUMMARY")

summaryTextView.text = orderSummary

}

}

**6. Add SharedPreferences for Saving Preferences (Optional)**

If you'd like to store whether a user wants cream and sugar for future orders:

kotlin

Copy code

// Save user preferences

val sharedPrefs = getSharedPreferences("CoffeeAppPrefs", MODE\_PRIVATE)

val editor = sharedPrefs.edit()

editor.putBoolean("hasCream", hasCream)

editor.putBoolean("hasSugar", hasSugar)

editor.apply()

// Load preferences on startup

val hasCream = sharedPrefs.getBoolean("hasCream", false)

val hasSugar = sharedPrefs.getBoolean("hasSugar", false)

creamCheckBox.isChecked = hasCream

sugarCheckBox.isChecked = hasSugar

**7. Running the App**

Once you’ve coded the above parts, run the app in Android Studio. It should allow users to input the quantity of coffee, select their preferences, and display the total cost in the order summary screen.

This case study showcases a simple yet complete app with two activities, basic UI interaction, and user input validation.

s