

Assignment 2

Alex Ngie Guan Ming

102765770

Task 1

Output

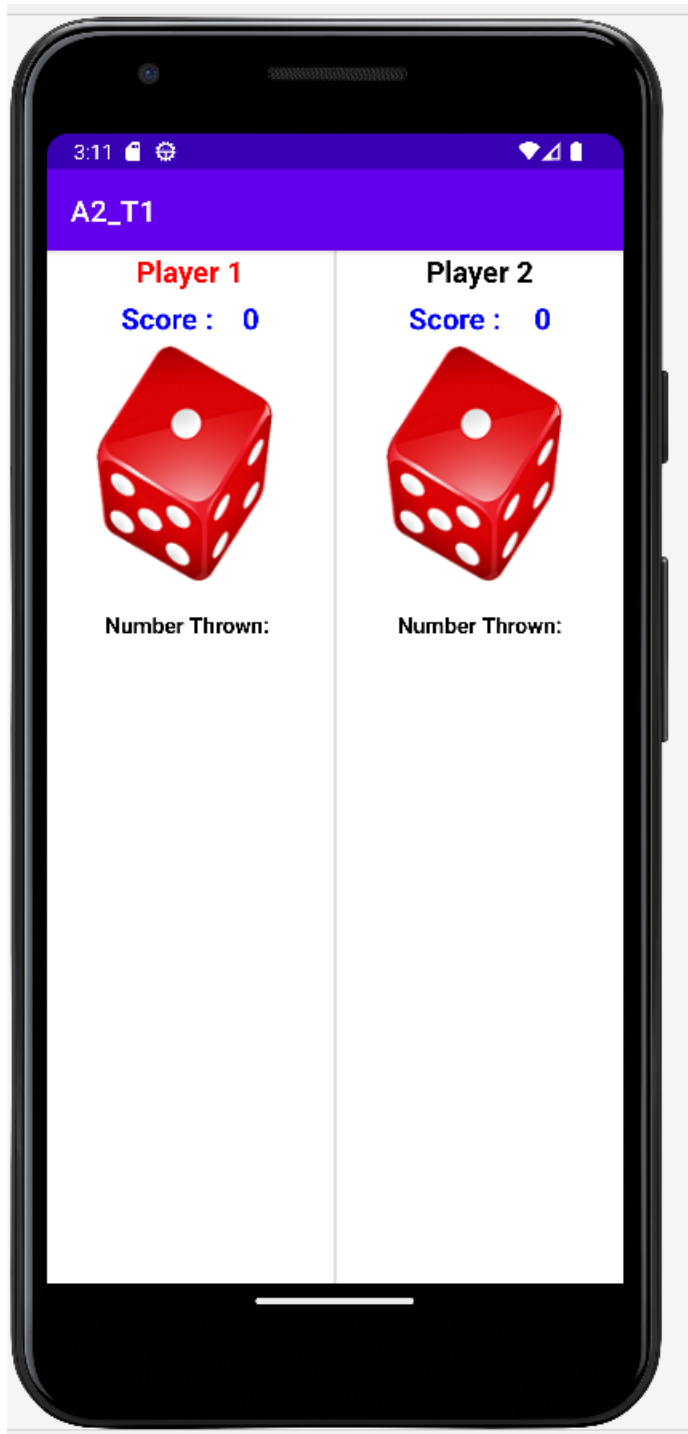


Figure 1 Start of App

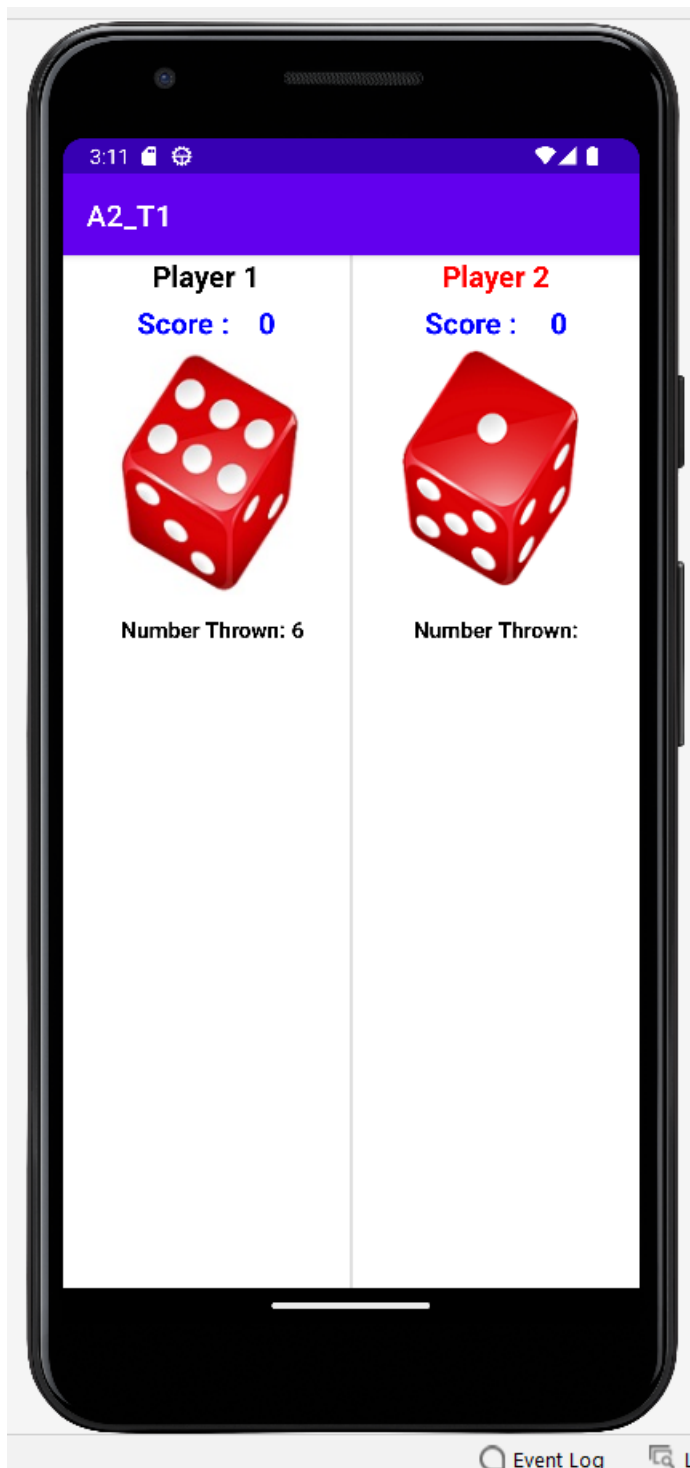


Figure 2 Player 1 throw, change player 2

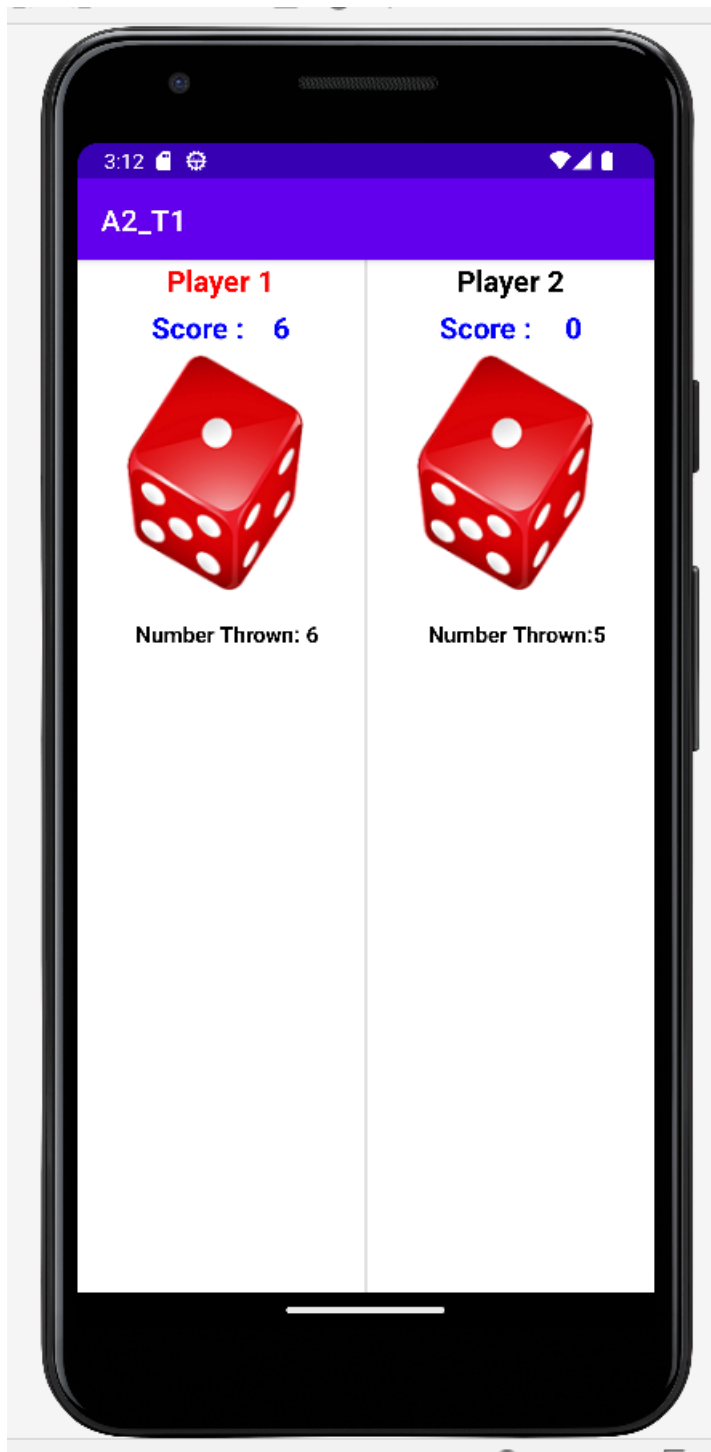


Figure 3 player 2 roll, player 1 won, score added



Figure 4 landscape view

Source code in Main.Activity.kt

```
package com.example.a2_t1

import android.content.res.Configuration
import android.graphics.Color
import android.graphics.drawable.DrawableWrapper
import android.media.Image
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.core.graphics.toColor
import org.w3c.dom.Text

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val img_player1: ImageView = findViewById(R.id.img_player1);
        val img_player2: ImageView = findViewById(R.id.img_player2);
        val txt2: TextView = findViewById(R.id.txt_throw2);
        val txt1: TextView = findViewById(R.id.txt_throw);
        var turn: Boolean = false;
        img_player1.setOnClickListener{
            if(!turn){

                val txt_p1 : TextView = findViewById(R.id.textView);
                val txt_p2 : TextView = findViewById(R.id.textView6);
                txt_p1.setTextColor(Color.parseColor("#000000"));
                txt_p2.setTextColor(Color.parseColor("#FF0000"))
                val num = (1..6).random();
                txt1.text = num.toString();
                img_player1.setImageResource(imageGenerate(num));
                turn = true;
            }
        };
        img_player2.setOnClickListener{
            if(turn){

                val txt_p1 : TextView = findViewById(R.id.textView);
                val txt_p2 : TextView = findViewById(R.id.textView6);
                txt_p2.setTextColor(Color.parseColor("#000000"));
                txt_p1.setTextColor(Color.parseColor("#FF0000"))
                val num = (1..6).random();
                txt2.text = num.toString();
                img_player2.setImageResource(imageGenerate(num));
                turn = false;
                reset(img_player1,img_player2,txt1,txt2);
            }
        }
    }

    override fun onConfigurationChanged(newConfig: Configuration) {
        super.onConfigurationChanged(newConfig)

        // Checks the orientation of the screen
        if (newConfig.orientation === Configuration.ORIENTATION_LANDSCAPE)
```

```

{
    Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show()
} else if (newConfig.orientation ===
Configuration.ORIENTATION_PORTRAIT) {
    Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show()
}
}

private fun imageGenerate(num:Int) : Int{
    val temp = when (num) {
        1 -> R.drawable.dice_one;
        2 -> R.drawable.dice_two;
        3 -> R.drawable.dice_three;
        4 -> R.drawable.dice_four;
        5 -> R.drawable.dice_five;
        6 -> R.drawable.dice_six;
        else -> { 0 }
    }
    return temp;
}

private fun
reset(p1:ImageView,p2:ImageView,p1throw:TextView,p2throw:TextView) {
    p1.setImageResource(R.drawable.dice_one);
    p2.setImageResource(R.drawable.dice_one);
    val p1score: TextView = findViewById(R.id.txt_score);
    val p2score: TextView = findViewById(R.id.txt_score2);
    var score1 = Integer.parseInt(p1score.text.toString());
    var score2 = Integer.parseInt(p2score.text.toString());
    var scorethrow1 = Integer.parseInt(p1throw.text.toString());
    var scorethrow2 = Integer.parseInt(p2throw.text.toString());
    if(scorethrow1 == scorethrow2 ){
        score1 += scorethrow1;
        score2 += scorethrow2;
    }else if(scorethrow1 > scorethrow2){
        score1 += scorethrow1;
    }else{
        score2 += scorethrow2;
    }
    p1score.text = score1.toString();
    p2score.text = score2.toString();
}
}

```

Source Code in activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/divider"
        android:textSize="20dp"
        android:textStyle="bold"
        android:textColor="@color/red"
        android:text="Player 1" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:textStyle="bold"
        android:textSize="20dp"
        android:textColor="@color/blue"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintBottom_toTopOf="@+id/img_player1"
        app:layout_constraintStart_toStartOf="@+id/img_player1"
        app:layout_constraintEnd_toStartOf="@+id/txt_score"
        android:text="Score :" />

    <TextView
        android:id="@+id/txt_score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="20dp"
        android:textColor="@color/blue"
        app:layout_constraintStart_toEndOf="@id/textView2"
        app:layout_constraintTop_toTopOf="@id/textView2"
        app:layout_constraintEnd_toEndOf="@+id/img_player1"
        android:text="0" />

    <ImageView
        android:id="@+id/img_player1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        app:layout_constraintTop_toBottomOf="@+id/textView2"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/divider"
        app:srcCompat="@drawable/dice_one" />

    <TextView
        android:id="@+id/txt_num_thrown"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        app:layout_constraintTop_toBottomOf="@+id/img_player1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/divider"
        android:text="0" />
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/img_player1"
        app:layout_constraintStart_toStartOf="@+id/img_player1"
        app:layout_constraintEnd_toEndOf="@+id/img_player1"
        android:layout_marginTop="10dp"
        android:textStyle="bold"
        android:textSize="15dp"
        android:textColor="@color/black"
        android:text="Number Thrown: " />

<TextView
    android:id="@+id/txt_throw"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toEndOf="@+id/txt_num_thrown"
    app:layout_constraintTop_toTopOf="@id/txt_num_thrown"
    app:layout_constraintBottom_toBottomOf="@+id/txt_num_thrown"
    android:textStyle="bold"
    android:textSize="15dp"
    android:textColor="@color/black"
    android:text="" />

<View
    android:id="@+id/divider"
    android:layout_width="2dp"
    android:layout_height="match_parent"
    android:background="?android:attr/listDivider"
    android:backgroundTint="#000000"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:textSize="20dp"
    android:textColor="@color/black"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toEndOf="@+id/divider"
    app:layout_constraintEnd_toEndOf="parent"
    android:text="Player 2" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:textStyle="bold"
    android:textSize="20dp"
    android:textColor="@color/blue"
    app:layout_constraintTop_toBottomOf="@+id/textView6"
    app:layout_constraintStart_toStartOf="@id/img_player2"
    app:layout_constraintEnd_toStartOf="@id/txt_score2"
    android:text="Score : " />

<TextView
    android:id="@+id/txt_score2"

```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:textSize="20dp"
    android:textColor="@color/blue"
    app:layout_constraintStart_toEndOf="@+id/textView7"
    app:layout_constraintTop_toTopOf="@id/textView7"
    app:layout_constraintBottom_toBottomOf="@id/textView7"
    app:layout_constraintEnd_toEndOf="@id/img_player2"
    android:text="0" />
```

```
<ImageView
    android:id="@+id/img_player2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    app:layout_constraintStart_toEndOf="@id/divider"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textView7"
    app:srcCompat="@drawable/dice_one" />
```

```
<TextView
    android:id="@+id/txt_num_thrown2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:textStyle="bold"
    android:textSize="15dp"
    android:textColor="@color/black"
    app:layout_constraintTop_toBottomOf="@+id/img_player2"
    app:layout_constraintStart_toStartOf="@id/img_player2"
    app:layout_constraintEnd_toEndOf="@+id/img_player2"
    android:text="Number Thrown:" />
```

```
<TextView
    android:id="@+id/txt_throw2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:textSize="15dp"
    android:textColor="@color/black"
    app:layout_constraintStart_toEndOf="@id/txt_num_thrown2"
    app:layout_constraintTop_toTopOf="@id/txt_num_thrown2"
    app:layout_constraintBottom_toBottomOf="@id/txt_num_thrown2"
    android:text="" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Source Code in AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.A2_T1"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"

android:configChanges="orientation|screenSize|screenLayout|keyboardHidden"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>

            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
            </activity>
        </application>

</manifest>
```

Task 2

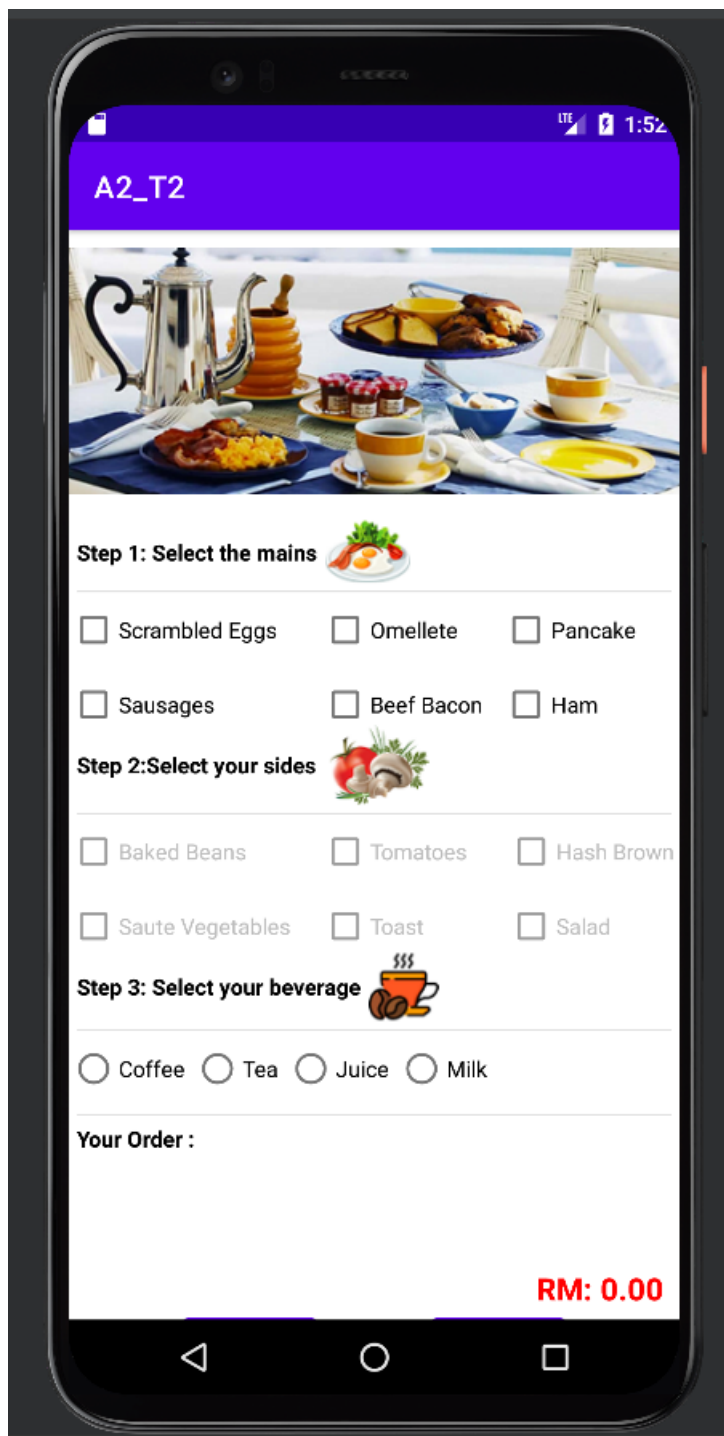


Figure 5 First Open App UI

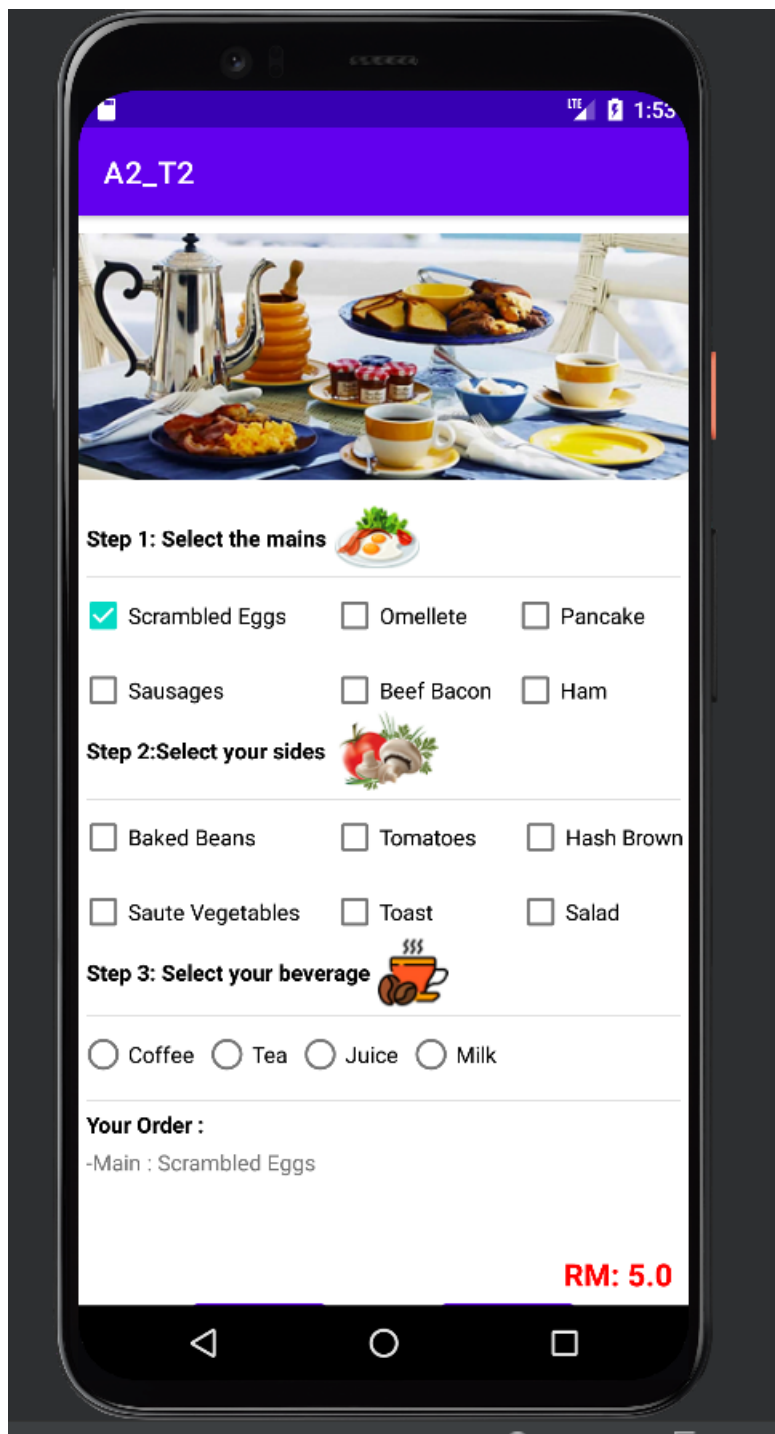


Figure 6 After Select At least One Main

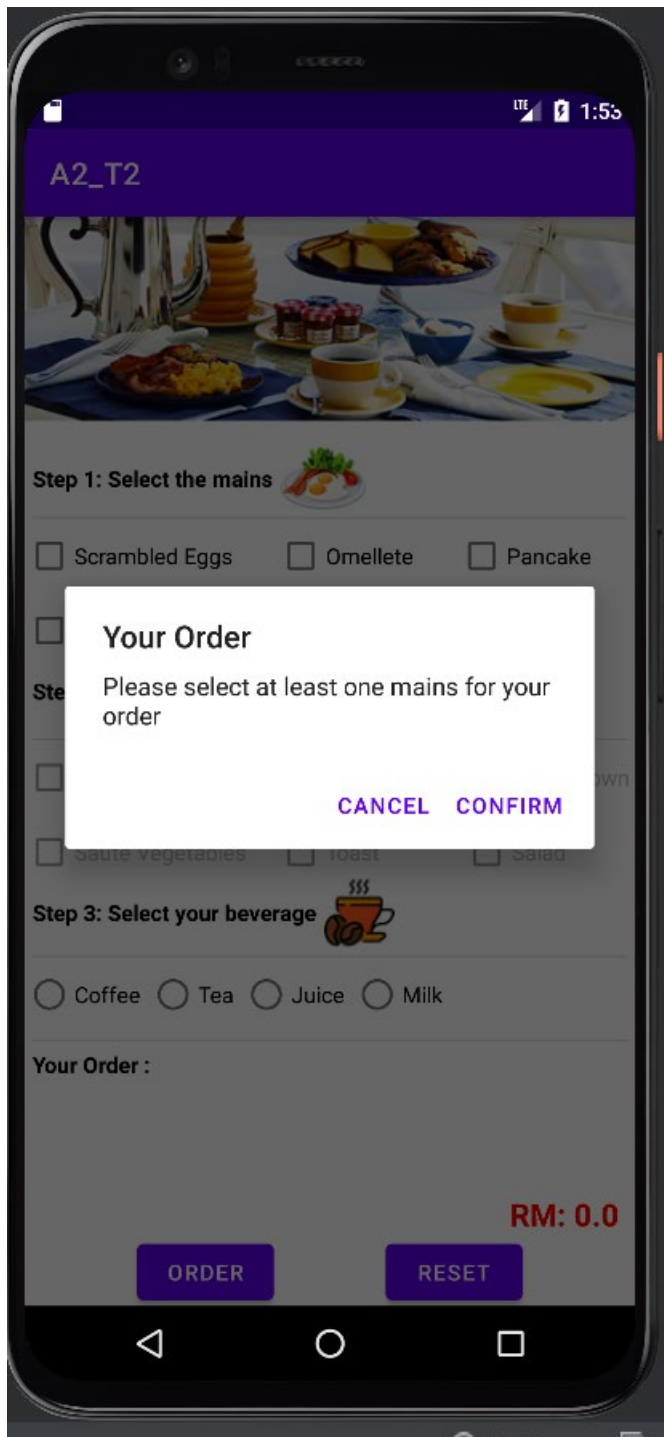


Figure 7 Error Message Without Select ANYthing

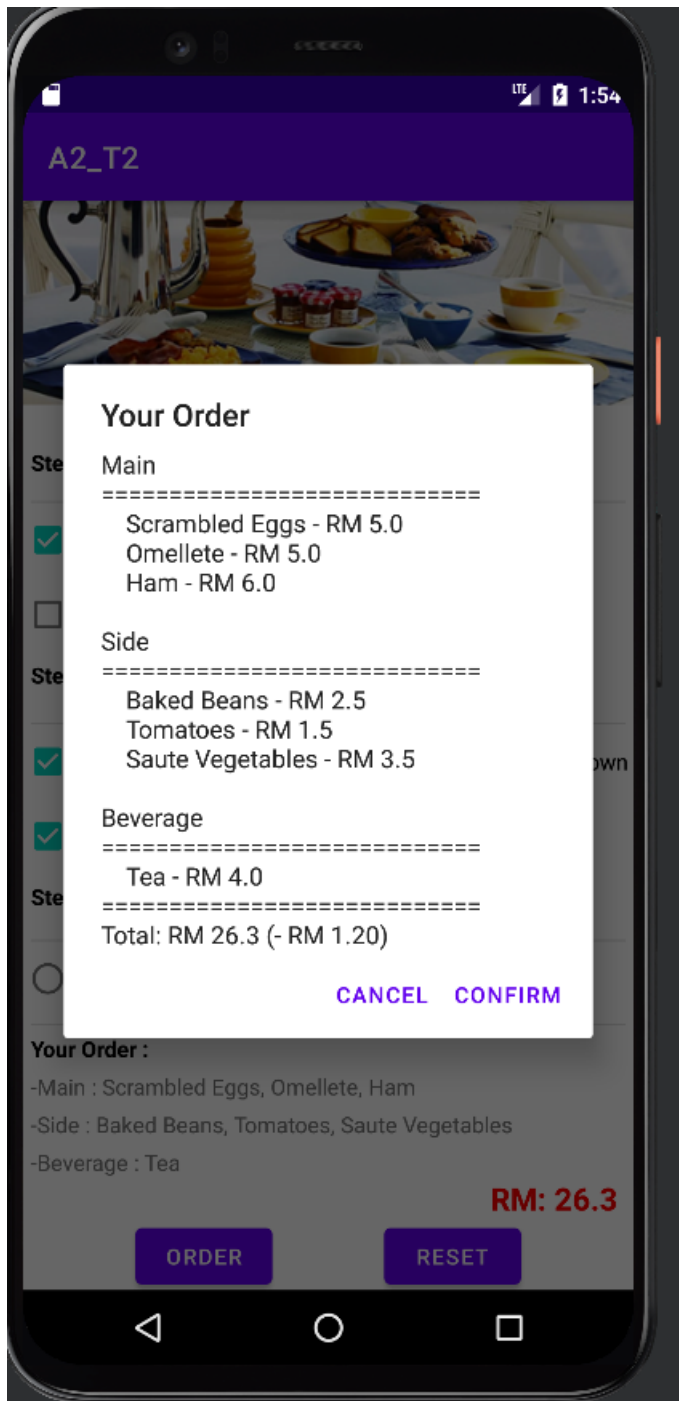
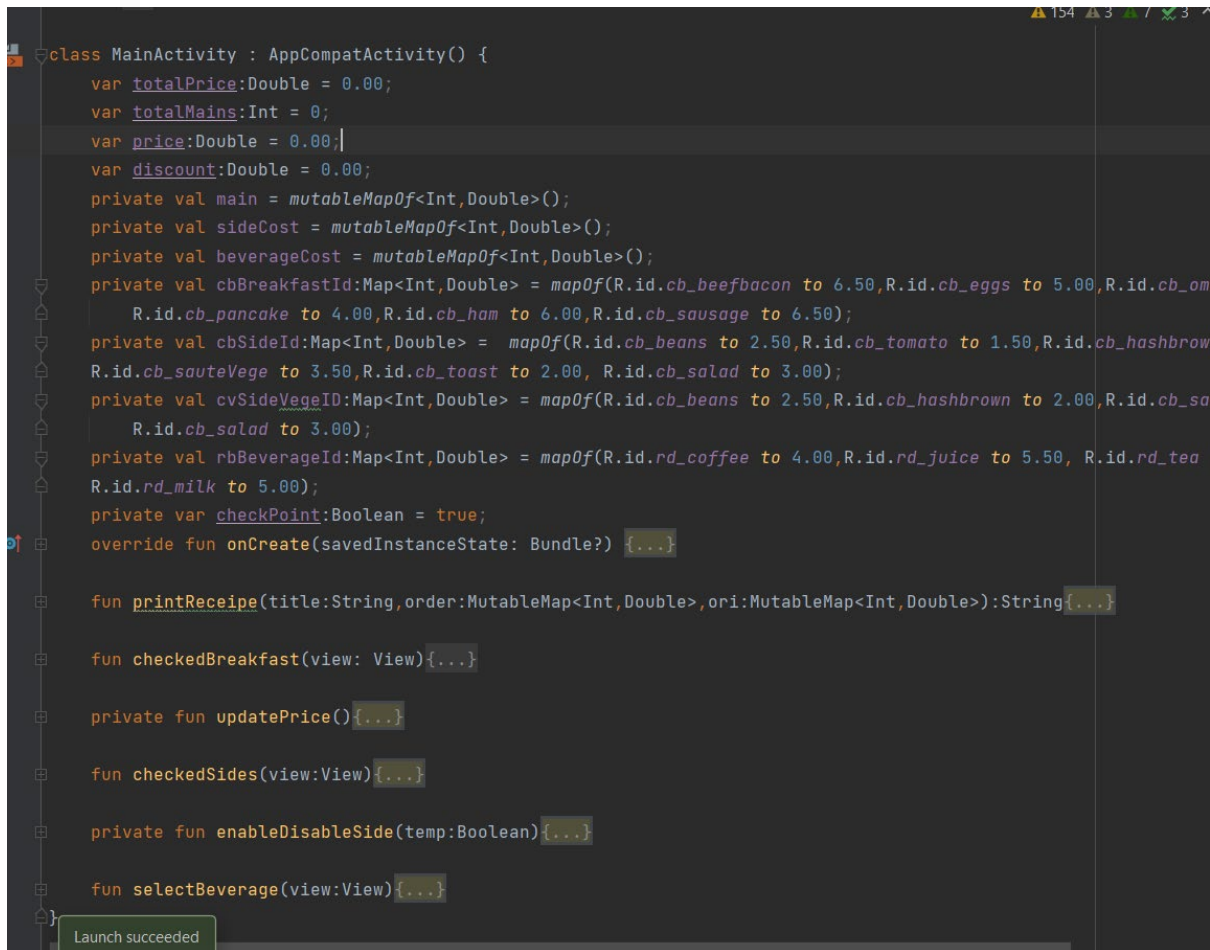


Figure 8 Receipt Printing



```

class MainActivity : AppCompatActivity() {
    var totalPrice:Double = 0.00;
    var totalMains:Int = 0;
    var price:Double = 0.00;
    var discount:Double = 0.00;
    private val main = mutableMapOf<Int,Double>();
    private val sideCost = mutableMapOf<Int,Double>();
    private val beverageCost = mutableMapOf<Int,Double>();
    private val cbBreakfastId:Map<Int,Double> = mapOf(R.id.cb_beefbacon to 6.50,R.id.cb_eggs to 5.00,R.id.cb_om
        R.id.cb_pancake to 4.00,R.id.cb_ham to 6.00,R.id.cb_sausage to 6.50);
    private val cbSideId:Map<Int,Double> = mapOf(R.id.cb_beans to 2.50,R.id.cb_tomato to 1.50,R.id.cb_hashbrow
        R.id.cb_sauteVege to 3.50,R.id.cb_toast to 2.00, R.id.cb_salad to 3.00);
    private val cvSideVegeID:Map<Int,Double> = mapOf(R.id.cb_beans to 2.50,R.id.cb_hashbrown to 2.00,R.id.cb_sa
        R.id.cb_salad to 3.00);
    private val rbBeverageId:Map<Int,Double> = mapOf(R.id.rd_coffee to 4.00,R.id.rd_juice to 5.50, R.id.rd_tea
        R.id.rd_milk to 5.00);
    private var checkpoint:Boolean = true;
    override fun onCreate(savedInstanceState: Bundle?) {...}

    fun printReceipt(title:String,order:MutableMap<Int,Double>,ori:MutableMap<Int,Double>):String{...}

    fun checkedBreakfast(view: View){...}

    private fun updatePrice(){...}

    fun checkedSides(view:View){...}

    private fun enableDisableSide(temp:Boolean){...}

    fun selectBeverage(view:View){...}
}

```

Launch succeeded

Figure 9 variable and function used

TotalPrice used to save total amount of user ordered in real time.

TotalMains used to calculate how many mains food user selected in real time

Price used to save the beverage price

Discount used to save total amount deducted, it depends on the user selection

mutableMapOf<Int,Double>

main, sideCost, beverageCost real time add and remove the item user selected and deselected, used in printing “Your Order” below and the receipt printing.

cbSideId, cbSideId, rbBeverageId stored all information of food in main, sides and drink – beverage.

cbSideVegeID stored all information of side that is category in vegetable.

checkpoint used to check when to enable and disable the sides checkbox. In initialize is true, once the totalMains become zero. It will set into false. Once user uncheck all mains, it will set into true again. This variable is used to prevent meaningless repeatedly enable side checkbox.

Function

printReceipt used to print the order information on the alert box, it returns type is string. This function also do the calculation of total amount of discount.

checkedBreakFast is an onclick function that declared in XML. It will be called once the checkbox that declared this function is checked or unchecked. This function is only declared in mains checkbox in XML. If **checked**, price of the main will total up into the total price, and the information will store into the mutableMapOf main. If **unchecked**, price will deducted from the main and information will removed from mutableMapOf main.

updatePrice is used to update the discounts of sides. It will real time updated the price of the side to user and the total amount. For example, when user **first selected 2 mains and 3 side** that is category as vegetable, after this selected 1 more item in mains, it will call the updatePrice function to update the price of side(discount 20%). Once the main item less than 2, all discounted side will back to normal price.

checkedSides is an onclick function that declared in XML. It will be called once the checkbox that declared this function is checked or unchecked. This function is only declared in side checkbox in XML. If **checked**, first will check the **totalMains is equal or more than 3**, yes all side category as vegetables will **discount 20%** and store information into the mutableMapOf sideCost. If **unchecked**, deducted the item price unchecked from totalPrice and remove information from mutableMapOf sideCost.

enableDisableSide used to enable and disable all checkbox of sides. If totalmains > 0 and checkpoint is true, it will call this function and enable all the checkbox and the checkpoint will set into false.

Now current is totalMains > 0 and !checkPoint, so it is not going to run enableDisableSide in this if else statement.

When user unchecked all mains, totalMains will became zero, now it fulfil the else statement, !checkpoint and totalMains == 0, checkpoint will set into true and disable all the side checkbox.

Because of this statement is contains in the function of checkedBreakFast. To prevent it run every time when user check or uncheck checkbox in main, it all relies on the checkpoint. Code provided at below.

```
if(totalMains > 0 && checkPoint){
    checkPoint = false;
    enableDisableSide(true);
}else if(!checkPoint && totalMains == 0){
    temp = "";
    checkPoint = true;
    enableDisableSide(false);
    sideCost.clear();
}
```

selectBeverage is an onclick function that declared in XML. It will be called once the radio button that declared this function is checked. This function is only declared in beverage radio button in XML. If **checked**, when **price is zero** and **mutableMapOf beverageCost is not empty**. Checked beverage price will store into the price and the beverage information will store into the mutableMapOf beverageCost. Otherwise, reassign the beverage price into **price** and information stored in **mutableMapOf beverageCost** will be clear and store the checked one beverage information.

Source Code MainActivity.kt

```
package com.example.a2_t2

import android.content.DialogInterface
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.provider.MediaStore.Audio.Radio
import android.view.View
import android.widget.Button
import android.widget.CheckBox
import android.widget.RadioButton
import android.widget.TextView
import androidx.appcompat.app.AlertDialog
import org.w3c.dom.Text
import java.text.DecimalFormat

class MainActivity : AppCompatActivity() {
    var totalPrice:Double = 0.00;
    var totalMains:Int = 0;
    var price:Double = 0.00;
    var discount:Double = 0.00;
    private val main = mutableMapOf<Int,Double>();
    private val sideCost = mutableMapOf<Int,Double>();
    private val beverageCost = mutableMapOf<Int,Double>();
    private val cbBreakfastId:Map<Int,Double> = mapOf(R.id.cb_beefbacon to
6.50,R.id.cb_eggs to 5.00,R.id.cb_omellete to 5.00,
        R.id.cb_pancake to 4.00,R.id.cb_ham to 6.00,R.id.cb_sausage to
6.50);
    private val cbSideId:Map<Int,Double> = mapOf(R.id.cb_beans to
2.50,R.id.cb_tomato to 1.50,R.id.cb_hashbrown to 2.00,
        R.id.cb_sauteVege to 3.50,R.id.cb_toast to 2.00, R.id.cb_salad to
3.00);
    private val cvSideVegeID:Map<Int,Double> = mapOf(R.id.cb_beans to
2.50,R.id.cb_hashbrown to 2.00,R.id.cb_sauteVege to 3.50,
        R.id.cb_salad to 3.00);
    private val rbBeverageId:Map<Int,Double> = mapOf(R.id.rd_coffee to
4.00,R.id.rd_juice to 5.50, R.id.rd_tea to 4.00,
        R.id.rd_milk to 5.00);
    private var checkPoint:Boolean = true;
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        val btnReset:Button = findViewById(R.id.btn_reset);
        val btnOrder:Button = findViewById(R.id.btn_order);
        val txtPrice :TextView = findViewById(R.id.txt_cost);
        val txtSummary:TextView = findViewById(R.id.txt_summary);
        val txtSide:TextView = findViewById(R.id.txt_side);
        val txtBeverage:TextView = findViewById(R.id.txt_beverage);
        var title:String = "";
        btnReset.setOnClickListener() {
            totalPrice = 0.00;
            totalMains = 0;
            discount = 0.00;
            checkPoint = true;
            sideCost.clear();
            main.clear();
            price = 0.00;
            for((key,value) in cbBreakfastId){
                val temp:CheckBox = findViewById(key);
                temp.isChecked = false;
```

```

    }
    for((key,value) in rbBeverageId){
        val temp:RadioButton = findViewById(key);
        temp.isChecked = false;
    }
    txtSummary.text = "";
    txtSide.text = "";
    txtBeverage.text = "";
    txtPrice.text = "0.00";
    enableDisableSide(false);
}
btnOrder.setOnClickListener(){
    val dialogBuilder = AlertDialog.Builder(this);
    var message:String = "";

    if(main.isNotEmpty()){
        title = "Main\n=====\\n";
        message += printReceipe(title,main, cbBreakfastId as
MutableMap<Int, Double>);
    }else{
        message = "Please select at least one mains for your
order\\n";
    }
    if (sideCost.isNotEmpty()){
        title = "\\nSide\\n=====\\n";
        message += printReceipe(title,sideCost,cbSideId as
MutableMap<Int, Double>);
    }
    if(beverageCost.isNotEmpty()){
        message += "\\nBeverage\\n=====\\n";
        for((key,value) in beverageCost){
            var temp:RadioButton = findViewById(key);
            message += "    " + temp.text.toString() + " - RM
$value\\n";
        }
    }
    if(main.isNotEmpty()){
        var d1 = String.format("%.2f",discount);
        message += "=====\\n";
        message += "Total: RM $totalPrice (- RM $d1)";
    }
    // set message of alert dialog
    dialogBuilder.setMessage(message)
    // if the dialog is cancelable
    .setCancelable(false)
    // positive button text and action
    .setPositiveButton("Confirm",
DialogInterface.OnClickListener {
        dialog, id -> finish()
    })
    // negative button text and action
    .setNegativeButton("Cancel",
DialogInterface.OnClickListener {
        dialog, id -> dialog.cancel()
    })
    // create dialog box
    val alert = dialogBuilder.create()
    // set title for alert dialog box
    alert.setTitle("Your Order")
    // show alert dialog
    alert.show()

```

```

    }
}

fun
printReceipe(title:String,order:MutableMap<Int,Double>,ori:MutableMap<Int,D
ouble>):String{
    var message:String = title;
    for((key,value) in order){
        var temp:CheckBox = findViewById(key);
        message += "      " + temp.text.toString() + " - RM
"+ori[key]+"\\n";
        discount += (ori[key]!! - order[key]!!);
    }
    return message;
}

fun checkedBreakfast(view: View){

    val txtPrice :TextView = findViewById(R.id.txt_cost);
    val txtSummary:TextView = findViewById(R.id.txt_summary);
    var temp:String = "-Main : ";
    if(view is CheckBox){
        val checked:Boolean = view.isChecked;
        for((key,value) in cbBreakfastId){
            if(view.id == key){
                if(checked){
                    totalPrice += value;
                    totalMains += 1;
                    main[key] = value;
                }else{
                    totalPrice -= value;
                    totalMains -= 1;
                    main.remove(key);
                }
            }
        }
    }
}

var n:Int = 0;

for((key,value ) in main){
    n++;
    val x:CheckBox = findViewById(key);
    temp += x.text;
    if(n < main.size){
        temp += ", ";
    }

}

if(totalMains > 0 && checkPoint){
    checkPoint = false;
    enableDisableSide(true);
}else if(!checkPoint && totalMains == 0){
    temp = "";
    checkPoint = true;
    enableDisableSide(false);
    sideCost.clear();
}

updatePrice();
txtPrice.text = totalPrice.toString();
txtSummary.text = temp;

```

```

    }

    private fun updatePrice(){
        for((key,value) in sideCost){
            var x:Boolean = cvSideVegeID.containsKey(key);
            if(x){ // exist in choosed side list
                if(totalMains >= 3){ // true means main choose equal or
more than 3
                    if(cvSideVegeID[key] == value){ // if both price equal
                        totalPrice -= value;
                        totalPrice += (value*0.8);
                        sideCost[key] = (value*0.8);
                    }
                }else{
                    if(cvSideVegeID[key] != value){
                        totalPrice -= value;
                        totalPrice += cvSideVegeID[key]!!;
                        sideCost[key] = cvSideVegeID[key]!!;
                    }
                }
            }
        }
    }

    fun checkedSides(view:View){
        if(view is CheckBox){
            val checked:Boolean = view.isChecked;
            val exist:Boolean = sideCost.containsKey(view.id);
            if(exist){
                totalPrice -= sideCost[view.id]!!;
                sideCost.remove(view.id);
            }else{
                for((ley,value) in cbSideId){
                    if(view.id == ley){
                        if(checked){
                            if(totalMains >= 3 && view.id != R.id.cb_toast
&& view.id != R.id.cb_tomato){
                                totalPrice += (value*0.8);
                                sideCost[ley] = (value*0.8);
                            }else{
                                totalPrice += value;
                                sideCost[ley] = value;
                            }
                        }
                    }
                }
            }
        }
    }

    val txtSide:TextView = findViewById(R.id.txt_side);
    val txtPrice :TextView = findViewById(R.id.txt_cost);
    var temp:String = "-Side : ";
    var n:Int = 0;
    for((key,value) in sideCost){
        n++;
        var x = findViewById<TextView>(key);
        temp += x.text;
        if(n != sideCost.size){
            temp += ", ";
        }
    }
}

```

```

        txtSide.text = temp;
        txtPrice.text = totalPrice.toString();
    }

    private fun enableDisableSide(temp:Boolean) {
        for((key,value) in cbSideId) {
            var x:CheckBox = findViewById(key);
            if(temp) {
                x.isEnabled = true;
            } else {
                x.isChecked = false;
                x.isEnabled = false;
                totalPrice = price;
            }
        }
        val txtSide:TextView = findViewById(R.id.txt_side);
        txtSide.text = "";
    }

    fun selectBeverage(view:View) {
        var x:Int = 0;
        if(view is RadioButton) {
            var checked:Boolean = view.isChecked;
            for((key,value) in rbBeverageId) {
                if(view.id == key) {
                    if(checked) {
                        if(price == 0.0) {
                            price = value;
                        }
                    } else {
                        totalPrice -= price;
                        price = value;
                    }
                }
                x = key;
                totalPrice += value;
            }
        }
    }

    val rd:RadioButton = findViewById(x);
    val txtPrice :TextView = findViewById(R.id.txt_cost);
    val txtBeverage:TextView = findViewById(R.id.txt_beverage);
    txtBeverage.text = "-Beverage : " + rd.text;
    txtPrice.text = totalPrice.toString();
    beverageCost.clear();
    beverageCost[x] = rbBeverageId[x]!!;
}
}

```

Source Code activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:scrollbars="vertical"
    tools:context=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:id="@+id/img_banner"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:srcCompat="@drawable/banner2" />

            <TextView
                android:id="@+id/txt_step1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:text="Step 1: Select the mains"
                android:textColor="@color/black"
                android:textStyle="bold"
                app:layout_constraintBottom_toBottomOf="@+id/img_breakfast"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="@+id/img_breakfast" />

            <ImageView
                android:id="@+id/img_breakfast"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                app:layout_constraintStart_toEndOf="@+id/txt_step1"
                app:layout_constraintTop_toBottomOf="@+id/img_banner"
                app:srcCompat="@drawable/breakfast_icon" />

            <View
                android:id="@+id/divider_1"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:layout_margin="5dp"
                android:background="?android:attr/listDivider"
                app:layout_constraintTop_toBottomOf="@+id/img_breakfast" />

            <CheckBox
                android:id="@+id/cb_eggs"
                android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:onClick="checkedBreakfast"
        android:text="Scrambled Eggs"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/divider_1" />
```

```
<CheckBox
    android:id="@+id/cb_omellete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="checkedBreakfast"
    android:text="Omellete"
    app:layout_constraintEnd_toStartOf="@+id/cb_pancake"
    app:layout_constraintStart_toEndOf="@+id/cb_eggs"
    app:layout_constraintTop_toTopOf="@+id/cb_eggs" />
```

```
<CheckBox
    android:id="@+id/cb_pancake"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="checkedBreakfast"
    android:text="Pancake"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/cb_omellete"
    app:layout_constraintTop_toTopOf="@+id/cb_omellete" />
```

```
<CheckBox
    android:id="@+id/cb_sausage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="checkedBreakfast"
    android:text="Sausages"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/cb_eggs" />
```

```
<CheckBox
    android:id="@+id/cb_beefbacon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="checkedBreakfast"
    android:text="Beef Bacon"
    app:layout_constraintStart_toStartOf="@+id/cb_omellete"
    app:layout_constraintTop_toBottomOf="@+id/cb_omellete" />
```

```
<CheckBox
    android:id="@+id/cb_ham"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="checkedBreakfast"
    android:text="Ham"
    app:layout_constraintStart_toStartOf="@+id/cb_pancake"
    app:layout_constraintTop_toBottomOf="@+id/cb_pancake" />
```

```
<TextView
    android:id="@+id/txt_step2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Step 2:Select your sides "
    android:textColor="@color/black"
    android:textStyle="bold"
```



```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/cb_sausage" />

<ImageView
    android:id="@+id/img_side"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintBottom_toBottomOf="@id/txt_step2"
    app:layout_constraintStart_toEndOf="@id/txt_step2"
    app:layout_constraintTop_toTopOf="@+id/txt_step2"
    app:srcCompat="@drawable/side_icon" />

<View
    android:id="@+id/divider2"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="5dp"
    android:background="?android:attr/listDivider"
    app:layout_constraintTop_toBottomOf="@+id/img_side" />

<CheckBox
    android:id="@+id/cb_beans"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Baked Beans"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/divider2" />

<CheckBox
    android:id="@+id/cb_tomato"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Tomatoes"
    app:layout_constraintBottom_toBottomOf="@id/cb_beans"
    app:layout_constraintStart_toStartOf="@+id/cb_beefbacon"
    app:layout_constraintTop_toTopOf="@id/cb_beans" />

<CheckBox
    android:id="@+id/cb_hashbrown"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Hash Brown"
    app:layout_constraintBottom_toBottomOf="@id/cb_tomato"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@id/cb_ham"
    app:layout_constraintTop_toTopOf="@id/cb_tomato" />

<CheckBox
    android:id="@+id/cb_sauteVege"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Saute Vegetables"

```

```

        app:layout_constraintStart_toStartOf="@id/cb_beans"
        app:layout_constraintTop_toBottomOf="@id/cb_beans" />

<CheckBox
    android:id="@+id/cb_toast"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Toast"
    app:layout_constraintStart_toStartOf="@id/cb_tomato"
    app:layout_constraintTop_toBottomOf="@id/cb_tomato" />

<CheckBox
    android:id="@+id/cb_salad"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:onClick="checkedSides"
    android:text="Salad"
    app:layout_constraintStart_toStartOf="@id/cb_hashbrown"
    app:layout_constraintTop_toBottomOf="@id/cb_hashbrown" />

<TextView
    android:id="@+id/txt_step3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Step 3: Select your beverage"
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/cb_sauteVege" />

<ImageView
    android:id="@+id/img_coffee"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintBottom_toBottomOf="@id/txt_step3"
    app:layout_constraintStart_toEndOf="@id/txt_step3"
    app:layout_constraintTop_toTopOf="@id/txt_step3"
    app:srcCompat="@drawable/coffee_icon" />

<View
    android:id="@+id/divider3"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="5dp"
    android:background="?android:attr/listDivider"

    app:layout_constraintTop_toBottomOf="@id/img_coffee" />

<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/divider3">

```

```

<RadioButton
    android:id="@+id/rd_coffee"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="selectBeverage"
    android:layout_marginRight="5dp"
    android:text="Coffee" />

<RadioButton
    android:id="@+id/rd_tea"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="selectBeverage"
    android:layout_marginRight="5dp"
    android:text="Tea" />

<RadioButton
    android:id="@+id/rd_juice"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="selectBeverage"
    android:layout_marginRight="5dp"
    android:layout_weight="1"
    android:text="Juice" />

<RadioButton
    android:id="@+id/rd_milk"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="selectBeverage"
    android:layout_weight="1"
    android:text="Milk" />
</RadioGroup>

<View
    android:id="@+id/divider4"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="5dp"
    android:background="?android:attr/listDivider"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/radioGroup1" />

<TextView
    android:id="@+id/txt_order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Your Order : "
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/divider4" />

<TextView
    android:id="@+id/txt_summary"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text=""
    app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintTop_toBottomOf="@+id/txt_order" />

<TextView
    android:id="@+id/txt_side"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txt_summary"
    android:text="" />

<TextView
    android:id="@+id/txt_beverage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintTop_toBottomOf="@id/txt_side"
    app:layout_constraintStart_toStartOf="parent"
    android:text="" />

<TextView
    android:id="@+id/txt_rm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="RM: "
    android:textColor="@color/red"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@id/txt_cost"
    app:layout_constraintEnd_toStartOf="@id/txt_cost"
    app:layout_constraintTop_toTopOf="@id/txt_cost" />

<TextView
    android:id="@+id/txt_cost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:text="0.00"
    android:textColor="@color/red"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@id/btn_reset"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/txt_beverage" />

<Button
    android:id="@+id/btn_order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ORDER"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@id/btn_reset"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/txt_cost" />

<Button
    android:id="@+id/btn_reset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="RESET"
    app:layout_constraintEnd_toEndOf="parent"

```

```
        app:layout_constraintStart_toEndOf="@id/btn_order"
        app:layout_constraintTop_toTopOf="@id/btn_order" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```