MOBILE WEB APPLICATION ASSIGNMENT

MWAD

Student Name: Thaspeeha Vahithu

Surname: Vahithu

Module: Mobile Web Application Development

Demonstration Link: https://youtu.be/PtYiGLvfhOc

Table of Contents

Chapter 1 Introduction **2**

Chapter 2 User Experience **3**

2.1 APP 1: Button Interactions Interface 3

2.2 APP 2: Button Checkbox Interface 4

Chapter 3 Back End Development **8**

3.1 APP 1: Button Interaction Back End 8

3.2 APP 2: Button Checkbox Back End 8

Chapter 4 AI Tools Used **10**

4.1 APP 1: Button Interactions App 10

4.2 APP 2: Button Checkbox App 10

Chapter 5 Reflection and Discussion **11**

Chapter 6 Conclusion **12**

References **13**

Back-End Code 14

Chapter 1: Introduction:

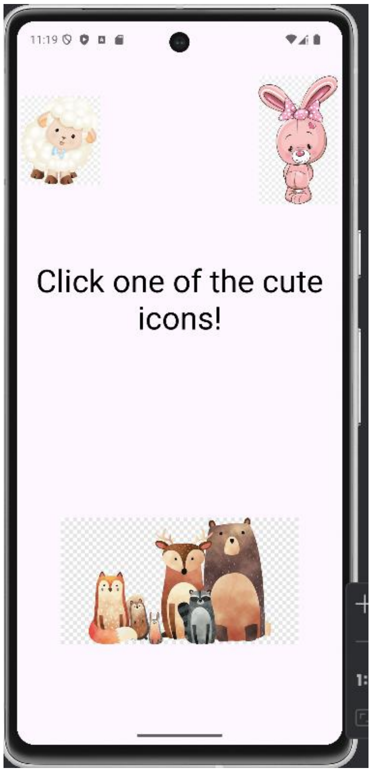
This Mobile Web Application report outlines the development process and functionality of five Android applications, with detailed designs and implementation. The task involved designing these applications using Android Studio, with Java selected as the back-end language, accompanied by video documentation explaining the detailed implementation of the applications. The first two applications are complete and bug-free, and these are demonstrated in the uploaded video. The following paragraph explains the content of each chapter:

Chapter 2 discusses the process of implementing user interfaces using Android Studio. Chapter 3 provides a critical analysis of the back-end development challenges and emphasizes the use of Java as the primary programming language. Chapter 4 discusses a critical reflection on the entire development process, highlighting lessons learned, and challenges faced. Finally, Chapter 5 summarizes the key outcomes of the project, emphasizing the strengths and limitations of the developed applications.

Chapter 2: User Experience:

2.1 APP 1: Button Interactions Interface

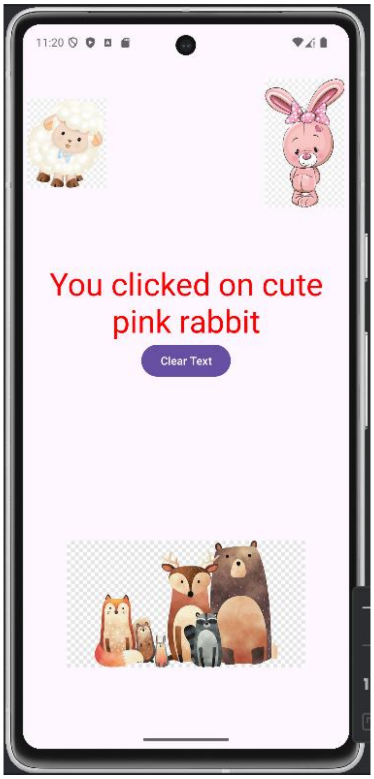
The first app is involved in developing a button interaction application featuring a *Next* button and four buttons positioned at each corner of the screen, set against a background image. When a button is selected, a text message is displayed. The Liner layout was chosen for the design as I was able to position by giving my own width and height values and the main reason for choosing the design was I was able to learn a lot of alignments in terms of match_parent and wrap_content and also linear layout was good in performance wise aligning greatly with the back end codes.



11:19

Click one of the cute icons!

Figure 1: The UI For APP 1



11:20

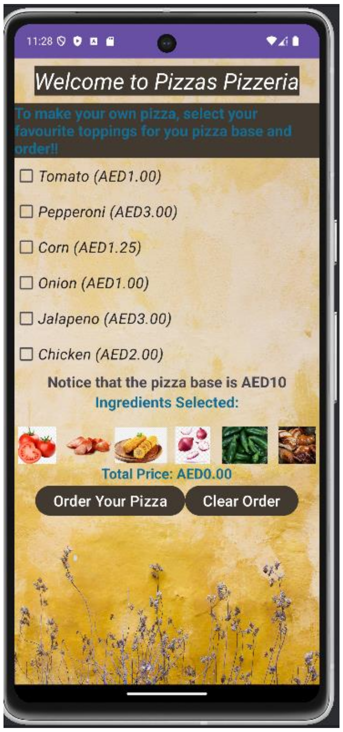
You clicked on cute pink rabbit

Figure 2: Shows text when button clicked

To enhance the app I changed the images for the application to a different image button design and this is done to not exactly copy what the requirements have asked us to do by giving the required images. As illustrated in Figures 1 and 2, the application was successfully implemented with a comfort design, as demonstrated in the accompanying video.

2.2 APP 2: Button Checkbox Interface

Image, Button, and Checkbox Interface consist of a linear layout that’s good performance-wise aligning greatly with the back-end codes. The app is involved in the implementation of Buttons and Checkboxes, where the selected Checkbox dynamically influences the options displayed as images. To provide context and functionality, the application was designed to simulate a food ordering system. Depending on the selected menu category via Checkboxes, the images display a corresponding list of items. Items selected from the checkboxes can then be submitted as orders, and a TextView field at the bottom of the screen displays the total price of the selected items. One topping can be ordered once only.


Welcome to Pizzas Pizzeria

To make your own pizza, select your favourite toppings for you pizza base and order!!

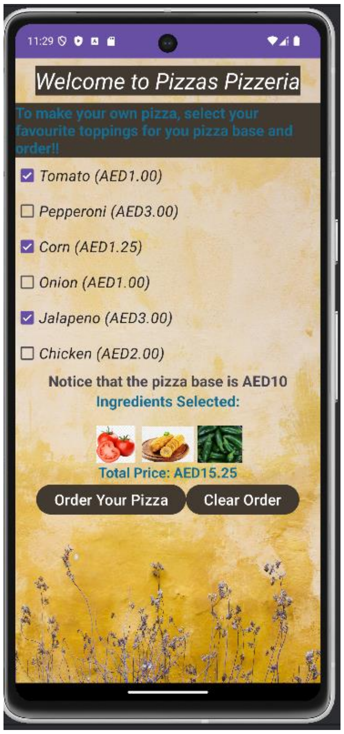
☐ Tomato (AED1.00)
☐ Pepperoni (AED3.00)
☐ Corn (AED1.25)
☐ Onion (AED1.00)
☐ Jalapeno (AED3.00)
☐ Chicken (AED2.00)

Notice that the pizza base is AED10

Ingredients Selected:

Total Price: AED0.00

Figure 3: The UI Of Checkbox Button Interaction


Welcome to Pizzas Pizzeria

To make your own pizza, select your favourite toppings for you pizza base and order!!

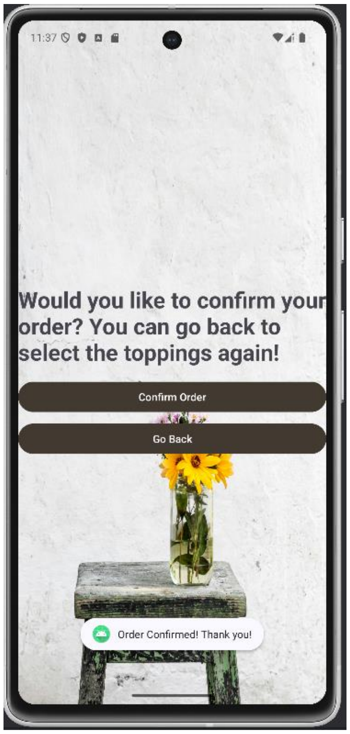
☑ Tomato (AED1.00)
☐ Pepperoni (AED3.00)
☑ Corn (AED1.25)
☐ Onion (AED1.00)
☑ Jalapeno (AED3.00)
☐ Chicken (AED2.00)

Notice that the pizza base is AED10

Ingredients Selected:

Total Price: AED15.25

Figure 4: User Input Added


Would you like to confirm your order? You can go back to select the toppings again!

Order Confirmed! Thank you!

Figure 5: Output of app

In an attempt to go above and beyond, it was decided to include a confirmation page with a toast message to show the user it’s been confirmed so this helps the user re-order if the user has changed his/her mind about changing the order. To enhance I have also added backgrounds to make the app classy in UI. This was not an integral part of the requirements but was applied as a minimal extra feature in the application. As illustrated in the above figures the application was successfully implemented with a comfort design, as demonstrated in the accompanying video.

Chapter 3: Back End development:

3.1 APP 1: Button Interaction Back End

Using an approach from ChatGPT for the design and back end I also faced some challenges with the design like orientation, width, etc values and I was able to solve the problem by understanding the logic behind it and I used some methods as well so it's easier for me to exercise the methods by calling them. The back-end logic for this task was straightforward. The back-end logic used if-else statements to differentiate between the buttons, determining the specific button that was pressed and executing a corresponding action. I had to overcome challenges in the visibility code for the images in the XML as well as in the back end in one of the methods (reset app).

The method *OnClickListener* simplifies the backend code by focusing solely on handling the actions triggered by each button press. Additionally, the image button required a separate handling method because it did not contain any text, necessitating custom logic to ensure the intended action. This method allowed for greater focus on refining user experience through the front-end design, while the back end could be kept concise and organized.

3.2 APP 2: Button Checkbox Back End

This task was more challenging compared to app 1, as it required learning how to implement checkboxes and radio buttons for the first time. Depending on the selected menu category via Checkboxes, the images display a corresponding list of items and a TextView field at the bottom of the screen displays the total price of the selected items following an approach recommended by Chatgpt. A topping is selected once only. This logic was implemented within an *onCheckedChanged* method, which functions similarly to *OnClickListener* allowing the checkbox function dynamically. I also used intent statements to select the toppings once click the order button it takes you to another page and confirms your order on that page also you can go back to re-order.

Overall, this experience contributed to developing problem-solving skills and a deeper understanding of adapting and refining code to meet specific requirements. The process of troubleshooting and modifying the solutions provided valuable insights into the integration of interactive UI elements.

Chapter 4 : AI Tools Used:

4.1 APP 1: Button Interactions App

ChatGPT referred me to the approach of using a linear layout performance-wise to align with the back end code, with visibility codes as ‘gone’ in the main activity itself to reduce complexity, and also an approach to creating method ‘HomePageReset’ which helps in calling where when you click one clear button resets to the main page and create method ‘ImageBrief’ which helps in calling where when you click one of the images, this is done to easily show the required text once called. It also referred me to use *OnClickListener* for the buttons.

4.2 APP 2: Button Checkbox App

ChatGPT referred me to the approach of using a linear layout and visibility codes in XML and back end to align well in the main activity itself to reduce complexity, and the approach to create methods ‘updatePricePreview' to add up when required toppings selected, ‘placeOrder’, ‘clearOrder’. An approach to use intent statements is given to redirect to the confirmation page once clicking the order button.

Chapter 5: Reflection & Discussion:

5.1. The successful development of the first two mobile applications while the last two applications was not completed well due to time constraints throughout this process. It became evident that a more advanced understanding of object-oriented programming (OOP), Relative Layout, and Frame Layout would have facilitated smoother progress had the task been approached later in the learning journey. Assistance from the platform ChatGPT proved invaluable for the last two applications, especially in the context of creating a HorizontalScrollView and using sharedPreferences respectively. For the first two applications, although the sample code was provided required to adapt it to fit the specific requirements of the project.

5.3: I have learned the importance of Object-Oriented Programming in Android Studio and also I have learned to understand the different types of logic and mathematical problem-solving and also and button interactions by creating methods and calling them.

Chapter 6: Conclusion:

My Mobile Web Application assignment as far now is the biggest project that I have completed I learned a lot through implementation especially the second app which required to use of problem-solving and mathematical logic and sparked my curiosity in implementing an app like this. To go above and beyond I enhanced by adding a confirmatory page, for future enhancements I would like to add sharedPreferences in this app which is a function to store confirmed order data.

My first app which is a cute icons app an easier and straightforward implementation. Since it is a cute icons app, for any future enhancements is adding animations and make it creative enough to make it entirely for kids to use. Even though this is way beyond comparing to my knowledge adding age restrictions can also enhance the app greatly.

References:

OpenAI (2023) 'Image Interactions', ChapGPT https://chatgpt.com/c/675d9e26-c10c-8013-bcb4-f2b7ca39d128 (LAST ACCESS 15 DEC-2024)

OpenAI (2023) 'Discussion on Pizza Ordering App', ChatGPT. https://chatgpt.com/c/673f0495-b3f4-8013-a163-aef3ab2e079b (LAST ACCESS 15 DEC-2024)

OPENAI (2023) ‘Register and Login’, ChatGPT. https://chatgpt.com/c/675bf206-9fe4-8013-aa6d-e777fc18d15a (LAST ACCESS 15 DEC-2024)

Back-End Code:

9.1 APP 1: CUTE ICONS APP

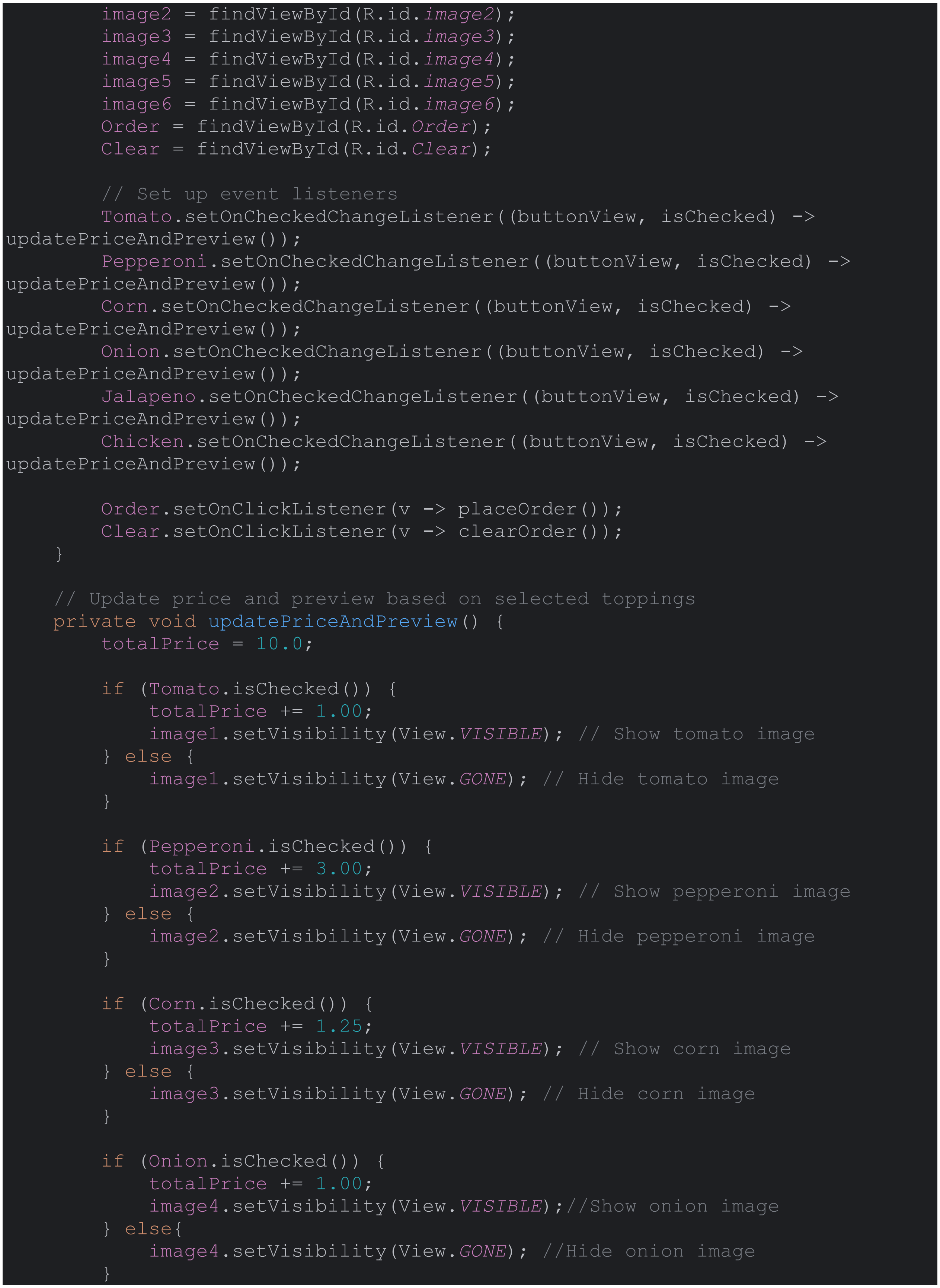
MainActivity.java

```
package com.example.cuteiconsapp;
//Statements imported
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

//Public class defined
public class MainActivity extends AppCompatActivity {
    //Define button controls with object
    ImageView sheep, rabbit, animals;
    TextView click;
    Button clear;

    @Override
    //From class: androidx. fragment. app. FragmentActivity Called when the
activity is starting.
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        //Bind the objects
        sheep = findViewById(R.id.sheep);
        rabbit = findViewById(R.id.rabbit);
        animals = findViewById(R.id.animals);
        click = findViewById(R.id.click);
        clear = findViewById(R.id.clear);

        //Setting functions to prepare action
        sheep.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                imageBrief("cute sheep", click, clear);  //Calling the method
                click.setTextColor(Color.CYAN);                  //Changes
text color
            }
        });

        rabbit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                imageBrief("cute pink rabbit", click, clear); //Calls the 
method
                click.setTextColor(Color.RED);
//Changes text color
            }
        });
        animals.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                imageBrief("cute animals", click, clear);      //Calls the 
method
                click.setTextColor(Color.BLUE);
//Changes text color
            }
        });

        clear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                homePageReset(click, clear);
//Calls the method
            }
        });
    }

        //method displays the name of the clicked image and shows the button
        private void imageBrief(String cuteIcon, TextView click, Button 
clear) {
          click.setText("You clicked on " + cuteIcon);                  // 
Display the name of the clicked image
          click.setVisibility(View.VISIBLE);                            // 
Make the TextView visible

          clear.setVisibility(View.VISIBLE);                            // 
Show the "Clear Text" button
    }

        //method resets the app's state to its initial condition
        private void homePageReset(TextView click, Button clear) {
          click.setText("Click one of the cute icons!");                // 
Reset the TextView text
          click.setTextColor(Color.BLACK);                              // 
Reset text color to black
          click.setVisibility(View.VISIBLE);                            // 
Resets the original "TextView"

          clear.setVisibility(View.GONE);                               // 
Hide the "Clear Text" button

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), 
(v, insets) -> {
            Insets systemBars = 
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, 
systemBars.bottom);
            return insets;
        });
    }
}
```

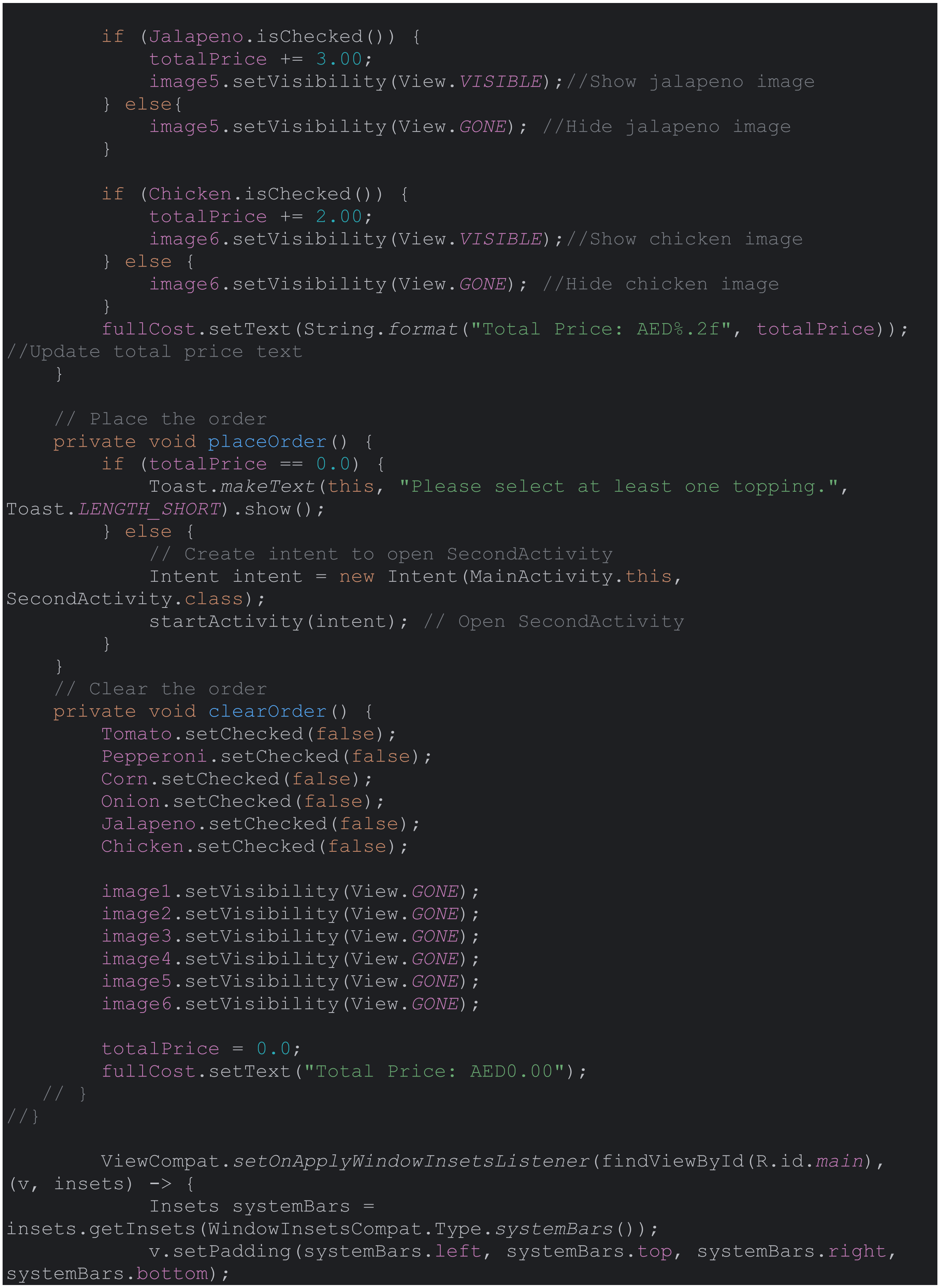
9.2 APP 2: PIZZA ORDERING APP

MainActivity.java

```
package com.example.pizzaorderingapp;
//Staments imported
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import org.w3c.dom.Text;

//Class defined
public class MainActivity extends AppCompatActivity {
    //Define the controls with objects
    CheckBox Tomato, Pepperoni, Corn, Onion, Jalapeno, Chicken;
    TextView fullCost;
    ImageView image1, image2, image3, image4, image5, image6 ;
    Button Order, Clear;

    //Initialize the cost
    private double totalPrice = 0.0;

    @Override
    //From class: androidx. fragment. app. FragmentActivity Called when the
activity is starting.
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize UI components to bind
        Tomato = findViewById(R.id.Tomato);
        Pepperoni = findViewById(R.id.Pepperoni);
        Corn = findViewById(R.id.Corn);
        Onion = findViewById(R.id.Onion);
        Jalapeno = findViewById(R.id.Jalapeno);
        Chicken = findViewById(R.id.Chicken);
        fullCost = findViewById(R.id.fullCost);
        image1 = findViewById(R.id.image1);
```



```
        image2 = findViewById(R.id.image2);
        image3 = findViewById(R.id.image3);
        image4 = findViewById(R.id.image4);
        image5 = findViewById(R.id.image5);
        image6 = findViewById(R.id.image6);
        Order = findViewById(R.id.Order);
        Clear = findViewById(R.id.Clear);

        // Set up event listeners
        Tomato.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());
        Pepperoni.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());
        Corn.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());
        Onion.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());
        Jalapeno.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());
        Chicken.setOnCheckedChangeListener((buttonView, isChecked) ->
updatePriceAndPreview());

        Order.setOnClickListener(v -> placeOrder());
        Clear.setOnClickListener(v -> clearOrder());
    }

    // Update price and preview based on selected toppings
    private void updatePriceAndPreview() {
        totalPrice = 10.0;

        if (Tomato.isChecked()) {
            totalPrice += 1.00;
            image1.setVisibility(View.VISIBLE); // Show tomato image
        } else {
            image1.setVisibility(View.GONE); // Hide tomato image
        }

        if (Pepperoni.isChecked()) {
            totalPrice += 3.00;
            image2.setVisibility(View.VISIBLE); // Show pepperoni image
        } else {
            image2.setVisibility(View.GONE); // Hide pepperoni image
        }

        if (Corn.isChecked()) {
            totalPrice += 1.25;
            image3.setVisibility(View.VISIBLE); // Show corn image
        } else {
            image3.setVisibility(View.GONE); // Hide corn image
        }

        if (Onion.isChecked()) {
            totalPrice += 1.00;
            image4.setVisibility(View.VISIBLE);//Show onion image
        } else{
            image4.setVisibility(View.GONE); //Hide onion image
        }
```



```
        if (Jalapeno.isChecked()) {
            totalPrice += 3.00;
            image5.setVisibility(View.VISIBLE);//Show jalapeno image
        } else{
            image5.setVisibility(View.GONE); //Hide jalapeno image
        }

        if (Chicken.isChecked()) {
            totalPrice += 2.00;
            image6.setVisibility(View.VISIBLE);//Show chicken image
        } else {
            image6.setVisibility(View.GONE); //Hide chicken image
        }
        fullCost.setText(String.format("Total Price: AED%.2f", totalPrice));
//Update total price text
    }

    // Place the order
    private void placeOrder() {
        if (totalPrice == 0.0) {
            Toast.makeText(this, "Please select at least one topping.",
Toast.LENGTH_SHORT).show();
        } else {
            // Create intent to open SecondActivity
            Intent intent = new Intent(MainActivity.this,
SecondActivity.class);
            startActivity(intent); // Open SecondActivity
        }
    }
    // Clear the order
    private void clearOrder() {
        Tomato.setChecked(false);
        Pepperoni.setChecked(false);
        Corn.setChecked(false);
        Onion.setChecked(false);
        Jalapeno.setChecked(false);
        Chicken.setChecked(false);

        image1.setVisibility(View.GONE);
        image2.setVisibility(View.GONE);
        image3.setVisibility(View.GONE);
        image4.setVisibility(View.GONE);
        image5.setVisibility(View.GONE);
        image6.setVisibility(View.GONE);

        totalPrice = 0.0;
        fullCost.setText("Total Price: AED0.00");
   // }
//}

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
```

```
            return insets;
        });
    }
}
```

SecondAcitvity.java

```
package com.example.pizzaorderingapp;
//Statements imported
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

//Class Defined
public class SecondActivity extends AppCompatActivity {
    //Define controls with objects
    Button confirmOrder, goBack;

    @Override
    //From class: androidx. fragment. app. FragmentActivity Called when the
activity is starting.
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_second);

        //Bind objects
        confirmOrder = findViewById(R.id.confirmOrder);
        goBack = findViewById(R.id.goBack);

        // Confirm Order Button
        confirmOrder.setOnClickListener(v -> {
            Toast.makeText(SecondActivity.this, "Order Confirmed! Thank
you!", Toast.LENGTH_LONG).show();
        });

        // Back to MainActivity Button
        goBack.setOnClickListener(v -> {
            Intent intent = new Intent(SecondActivity.this,
MainActivity.class);
            startActivity(intent);
            finish(); // Close SecondActivity
        });

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

```
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
    }
    }
```