Menu

- In android, **Menu** is a part of the user interface (UI) component which is used to handle some common functionality around the application.
- Types

1. Android Options Menu

   In android, **Options Menu** is a primary collection of menu items for an activity and it is useful to implement actions that have a global impact on the app, such as Settings, Search, etc.

2. Android Context Menu
   - In android, **Context Menu** is a floating menu that appears when the user performs a long click on an element and it is useful to implement actions that affect the selected content or context frame.
3. Android Popup Menu
   - In android, **Popup Menu** displays a list of items in a vertical list that's anchored to the view that invoked the menu and it's useful for providing an overflow of actions that related to specific content.

Define an Android Menu in XML File

- For all menu types, Android provides a standard XML format to define menu items.
- Instead of building a menu in our activity's code, we should define a menu and all its items in an XML menu resource and load menu resource as a Menu object in
  our activity or fragment.
- In android, to define menu, we need to create a new folder **menu** inside of our project resource directory (**res/menu/**) and add a new XML file to build the menu with the different elements.

Elements of menu resource file

Element	Description
<menu>	It's a root element to define a Menu in XML file and it will hold one or more and elements.
<item>	It is used to create a menu item and it represents a single item on the menu. This element may contain a nested <menu> element in order to create a submenu.
<group>	It's an optional and invisible for <item> elements. It is used to categorize the menu items so they share properties such as active state and visibility.

The **<item>** element in **menu** supports different type of attributes to define item's behaviour and appearance.

Following are the some of commonly used **<item>** attributes in android applications.

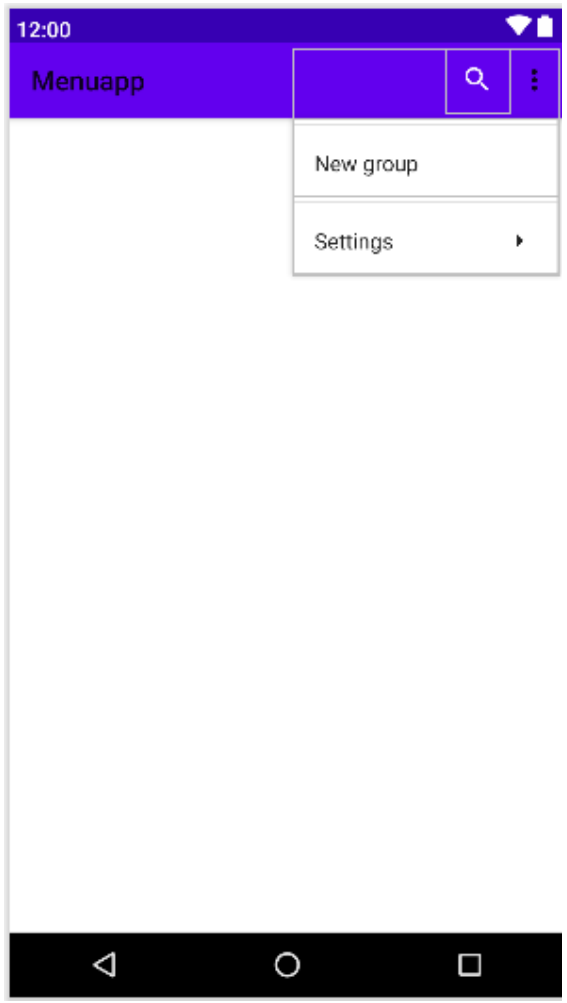
Attribute	Description
android: id	It is used to uniquely identify an element in the application.
android:icon	It is used to set the item's icon from drawable folder.
android: title	It is used to set the item's title
android:showAsAction	It is used to specify how the item should appear as an action item in the app bar.

Load Android Menu from an Activity

- Once we are done with creation of menu, we need to load the menu resource from our activity using **MenuInflater.inflate()**
- **MenuInflater** class is used to **instantiate menu XML files into Menu objects**.

Option menu example

Design a option menu (use whatsapp option menu as reference)



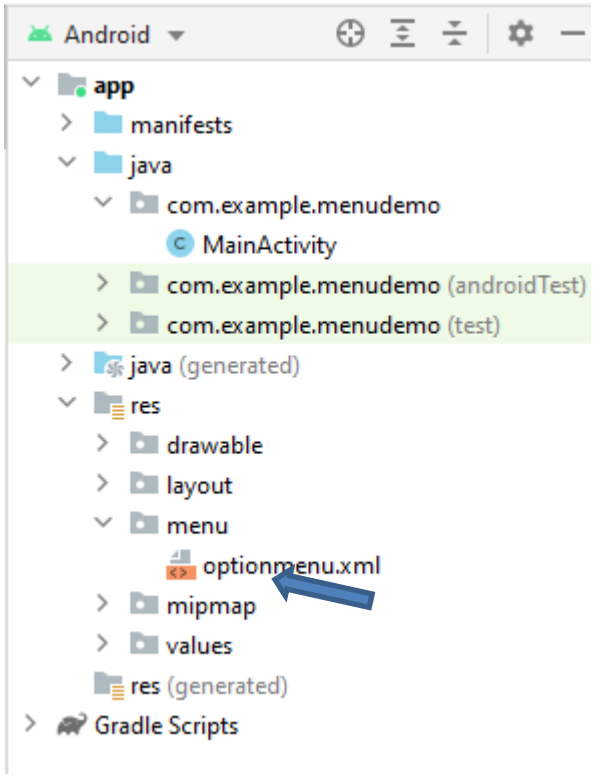
activity_main.xml

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

- Create resource directory named menu in res folder
- Create option menu resource(optionmenu.xml) in resource directory named menu under res folder

optionmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto" >
<item android:id="@+id/search"
   android:title="Search"
   android:icon="@drawable/ic_baseline_search_24"
   app:showAsAction="ifRoom"/>
   <item android:id="@+id/group"
       android:title="New group"/>
   <item android:id="@+id/settings"
       android:title="Settings">
       <menu>
           <item android:id="@+id/account"
               android:title="Account"/>
           <item android:id="@+id/chats"
               android:title="Chats"/>
           <item android:id="@+id/notifications"
               android:title="notifications"/>
       </menu>
   </item>
</menu>
```



MainActivity.java

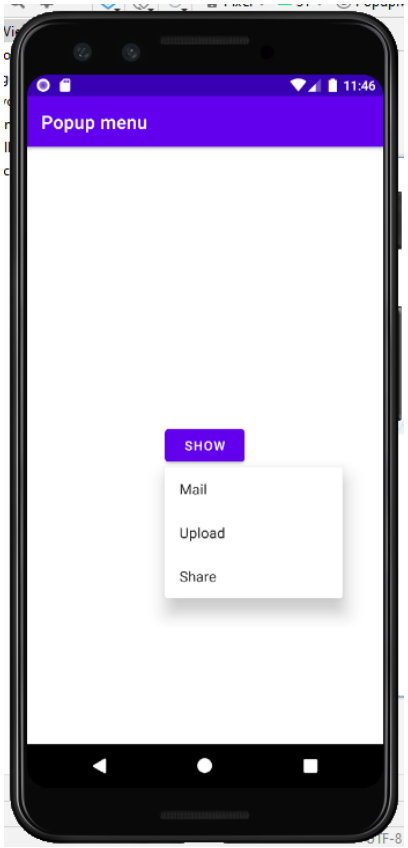
```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater=getMenuInflater();
        inflater.inflate(R.menu.optionmenu,menu);
        return true;
    }
```

MainActivity.java contd..

```
@Override
    public boolean onOptionsItemSelected(MenuItem item) {

        int itemId = item.getItemId();
        if (itemId == R.id.search) {
            Toast.makeText(getApplicationContext(), "Search clicked",
Toast.LENGTH_LONG).show();
        } else if (itemId == R.id.group) {
            Toast.makeText(getApplicationContext(), "New Group clicked",
Toast.LENGTH_LONG).show();
        } else if (itemId == R.id.account) {
            Toast.makeText(getApplicationContext(), "Account clicked",
Toast.LENGTH_LONG).show();
        } else if (itemId == R.id.chats) {
            Toast.makeText(getApplicationContext(), "Chats clicked", Toast.LENGTH_LONG).show();
        }
        else if (itemId == R.id.notifications) {
            Toast.makeText(getApplicationContext(), "Notifications clicked",
Toast.LENGTH_LONG).show();
        }
        return super.onOptionsItemSelected(item);
    }
}
```

Popup menu example

- Create an application which has a button and displays popup menu when user clicks that button.



popupmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/r
es/android">
    <item android:id="@+id/mail"
        android:title="Mail" />
    <item android:id="@+id/upload"
        android:title="Upload" />
    <item android:id="@+id/share"
        android:title="Share" />
</menu>
```

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">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnShow"
        android:text="Show"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.PopupMenu;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PopupMenu popup=new PopupMenu(MainActivity.this,view);
                popup.inflate(R.menu.popup_menu);
                popup.show();
                popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                    @Override
                    public boolean onMenuItemClick(MenuItem menuItem) {
                        Toast.makeText(getApplicationContext(), "Selected Item: " + menuItem.getTitle(), Toast.LENGTH_SHORT).show();
                        return true;
                    }
                });
            }
        });
    }
}
```

Context menu

- **Design an application which has Image and display context menu on that image and also create and redirect to different activities.**

contextmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/Open_new" android:title="Open in new
Activity"/>
   <item android:title="Cancel" android:id="@+id/cancel"/>
</menu>
```

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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:scaleType="fitCenter"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/rose" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_image.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
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=".ImageActivity">
   <ImageView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:scaleType="fitXY"
       android:src="@drawable/rose"

app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.0"

app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toTopOf="parent"
       app:layout_constraintVertical_bias="0.0"
       tools:ignore="MissingConstraints" />

 <TextView
       android:id="@+id/textView"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Flower Image"
       android:textColor="@color/white"
       android:textSize="30dp"
       android:layout_marginTop="20dp"

android:layout_gravity="center_horizontal"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.147"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.058" />
</FrameLayout>
```

MainActivity.java

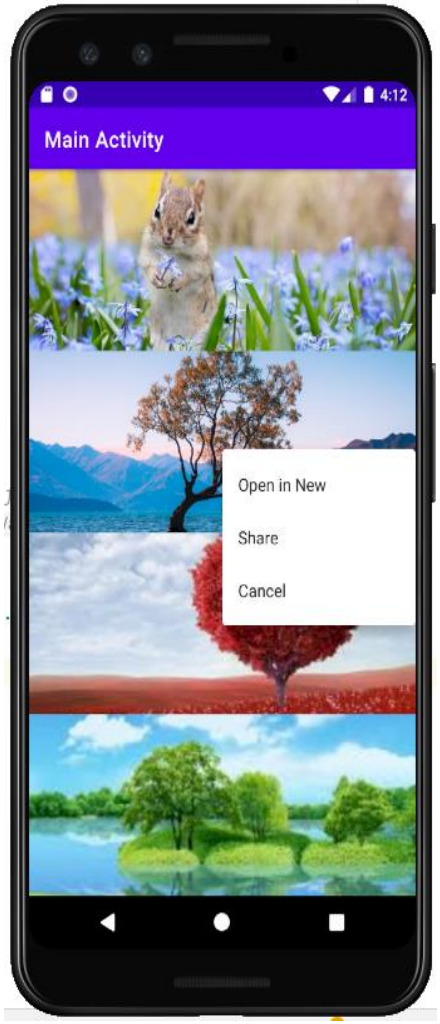
```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        registerForContextMenu(findViewById(R.id.imageView));
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
        MenuInflater inflater=getMenuInflater();
        inflater.inflate(R.menu.contextmenu,menu);
    }
```

MainActivity.java contd..

```
@Override
    public boolean onContextItemSelected(@NonNull MenuItem item) {
        int itemId = item.getItemId();
        if (itemId == R.id.open_new) {
            Intent i = new Intent(MainActivity.this, MainActivity2.class);
            startActivity(i);
        } else if (itemId == R.id.cancel) {
            Toast.makeText(getApplicationContext(), "Cancel clicked",
Toast.LENGTH_LONG).show();
        }
        return super.onContextItemSelected(item);
    }
}
```

Context Menu example

- **Design an application which has Images and display context menu on that image and also create and redirect to different activities.**



styles.xml: - File in values resource directory to apply common properties to image views

style.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <style name="image_props">
       <item
name="android:layout_width">match_parent</it
em>
       <item
name="android:layout_height">0dp</item>
       <item
name="android:layout_weight">1</item>
       <item
name="android:scaleType">centerCrop</item>
   </style>

</resources>
```

contextmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
>
    <item
        android:id="@+id/open_new"
        android:title="Open in New" />
    <item
        android:id="@+id/share"
        android:title="Share" />
    <item
        android:id="@+id/cancel"
        android:title="Cancel" />

</menu>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/nature"
        style="@style/image_props"
        android:tag="nature"
        android:src="@drawable/nature" />
    <ImageView
        android:id="@+id/nature1"
        style="@style/image_props"
        android:tag="nature1"
        android:src="@drawable/nature1" />
    <ImageView
        android:id="@+id/natue2"
        style="@style/image_props"
        android:tag="nature2"
        android:src="@drawable/nature2" />
    <ImageView
        android:id="@+id/nature3"
        style="@style/image_props"
        android:tag="nature3"
        android:src="@drawable/nature3" />

</LinearLayout>
```

MainActivity.java

```
package com.example.contextmenu;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.ContentInfo;
import android.view.ContextMenu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.os.PersistableBundle;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    ImageView []image;
    int selected_id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // changing titles
        getSupportActionBar().setTitle("Main Activity");
        image = new ImageView[4];
        int []ids = { R.id.nature, R.id.nature1, R.id.natue2, R.id.nature3 };
        for(int i = 0; i < 4; i++) {
            image[i] = findViewById(ids[i]);
            // registering images for context menu
            registerForContextMenu(image[i]);
        }
    }
```

MainActivity.java contd..

```
@RequiresApi(api = Build.VERSION_CODES.Q)
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.contextmenu, menu);
        // get the selected image drawable id
        selected_id = getResources().getIdentifier(v.getTag().toString(),
"drawable", this.getPackageName());

    }
    // overridden method to perform action on item selection
    // in our case we are passing image id to new activity
    @Override
    public boolean onContextItemSelected(@NonNull MenuItem item) {

        switch(item.getItemId()) {
            case R.id.open_new:
                Intent intent = new Intent(this, SecondActivity.class);
                intent.putExtra("id", selected_id);
                startActivity(intent);
                break;
            case R.id.cancel:
                Toast.makeText(this, "Cancel Clicked!",
Toast.LENGTH_SHORT).show();
                break;
        }
        return super.onContextItemSelected(item);
    }
```

activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".SecondActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="32dp"
        android:gravity="center"
        android:textColor="@color/teal_700"
        android:textSize="26sp"
        android:text="Image from another activity" />

    <!--    image view to display result image    -->
    <ImageView
        android:id="@+id/imgSeleced"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitCenter" />

</RelativeLayout>
```

SecondActivity.java

```
package com.example.contextmenu;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.Toast;

public class SecondActivity extends AppCompatActivity {
    ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        // changing title
        getSupportActionBar().setTitle("New Activity");

        // get the id of drawable from intent and setting source for
imageview
        int image_id = getIntent().getIntExtra("id", -1);
        imageView = findViewById(R.id.imgSeleced);
        imageView.setImageResource(image_id);
        Toast.makeText(this, "Opened in new activity..."+image_id,
Toast.LENGTH_SHORT).show();

    }
}
```