



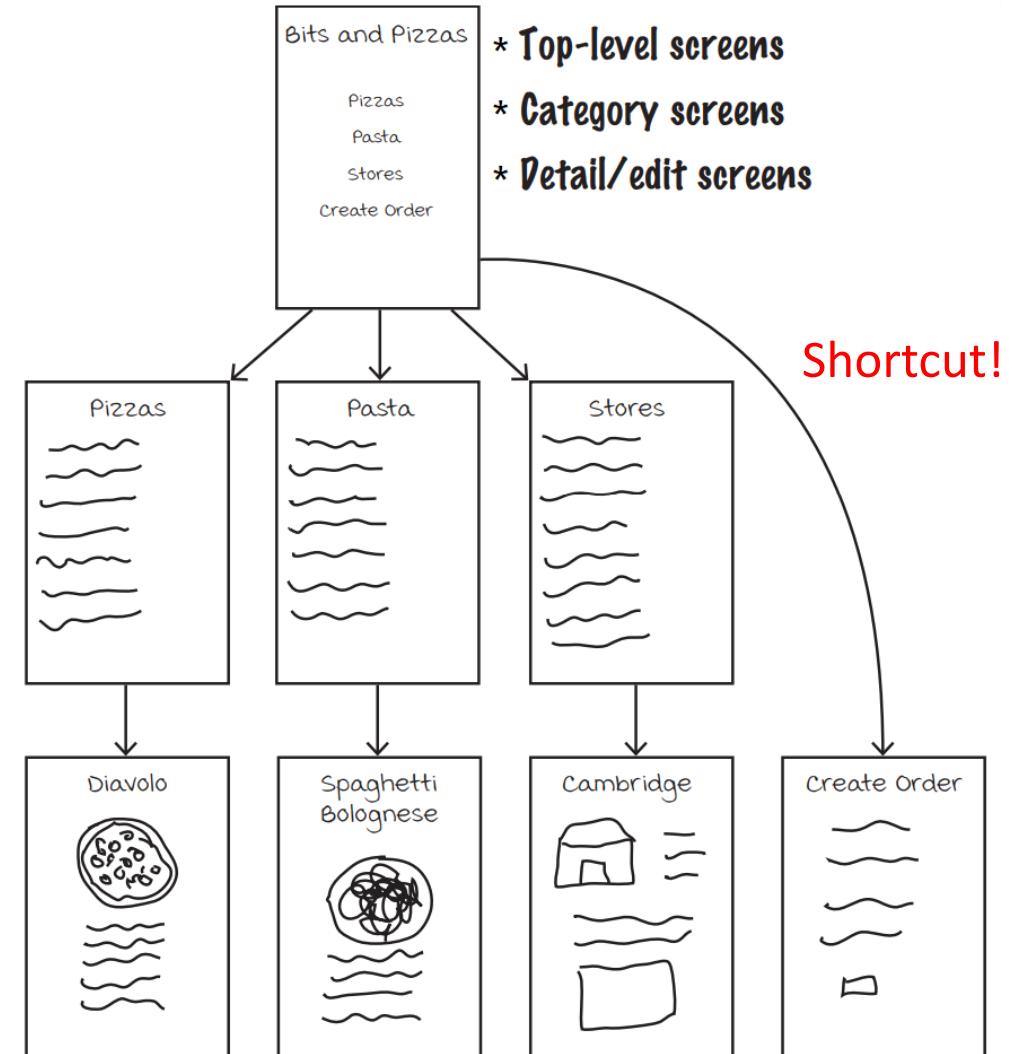
# Lab 4 – Navigation Bar

KUAN-TING LAI

2022/10/3

# APP Design Example

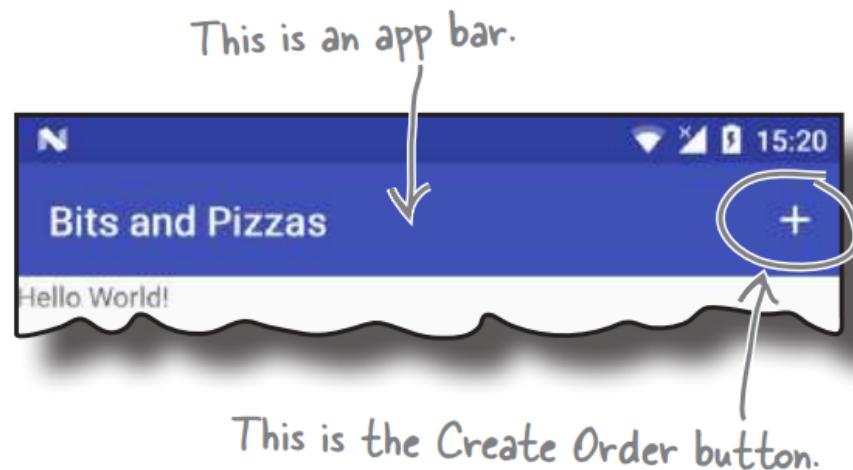
- General APP Hierarchy
  - Top -> Category -> Details
- Need a shortcut to important activity
- Pizza APP example



# APP Bar

---

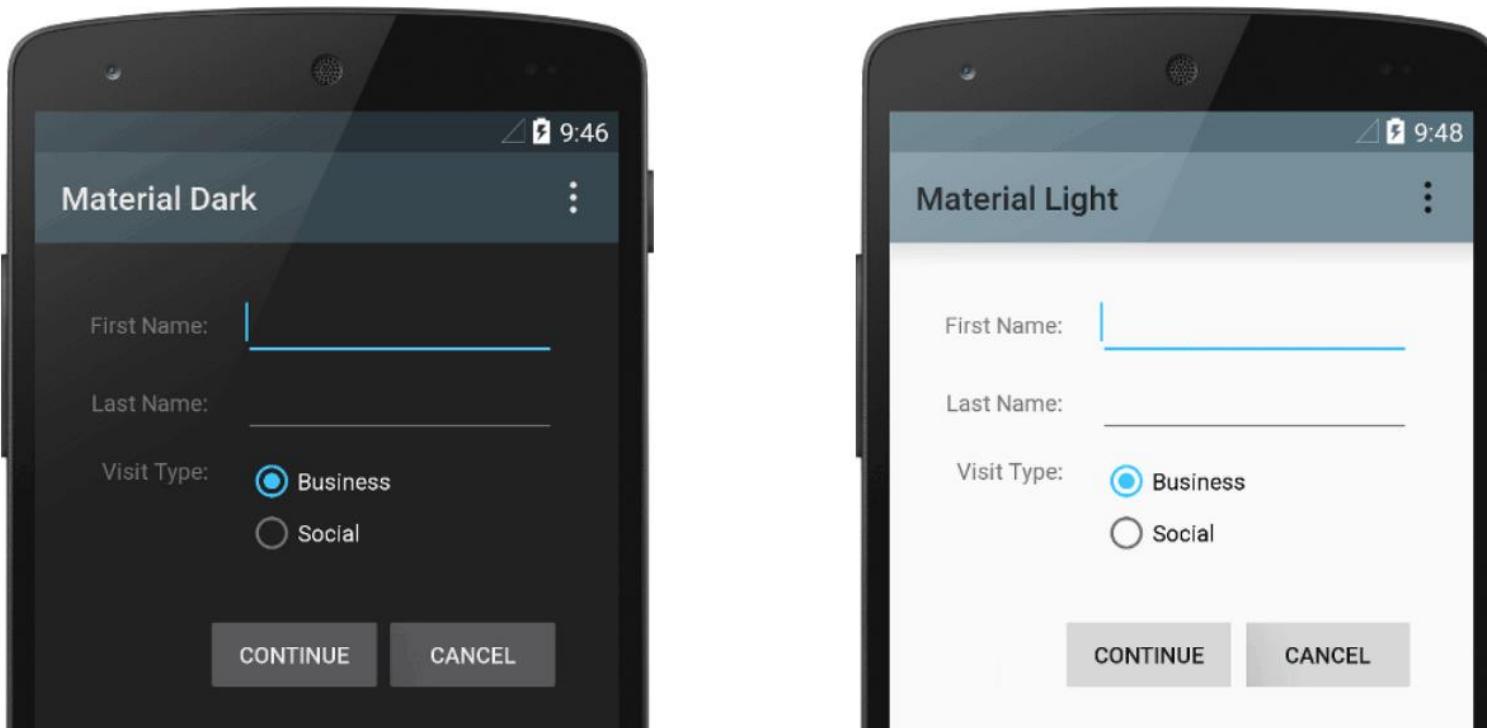
- Use APP bar (navigation bar) to provide shortcut



# Styles and Themes

---

- In order to support the dark theme functionality, make sure you are depending of the **Material Android library**.



Note :

Material Design Components (MDC) → For Jetpack (AndroidX)

# Change the Style of APP Bar

---

- Modify AndroidManifest.xml
  - Apply to the whole app

```
<application  
    ...  
        android:theme="@style/Theme.MaterialComponents.DayNight.DarkActionBar"  
    ...  
</application>
```

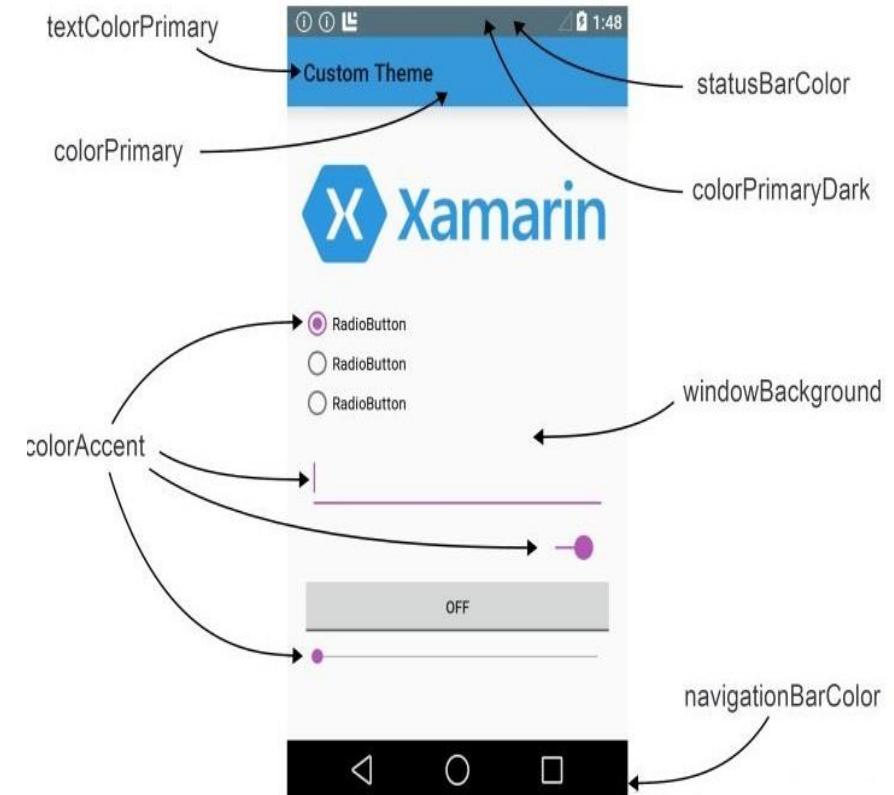
- Apply to just one activity

```
<application  
    ...  
        <activity android:theme="@style/Theme.MaterialComponents.DayNight.DarkActionBar"  
    ...  
</application>
```

# Customize Theme

- Override items in app/res/values/themes/themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.MyApplication"
        parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```



Note :

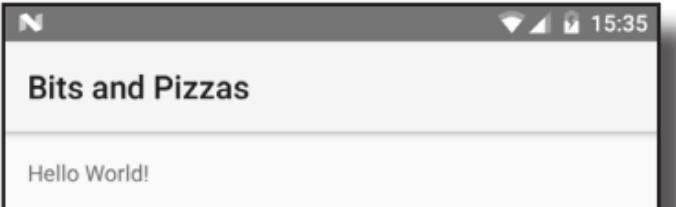
Theme.AppCompat : for above API 7+

Theme.Material : for above API 21+

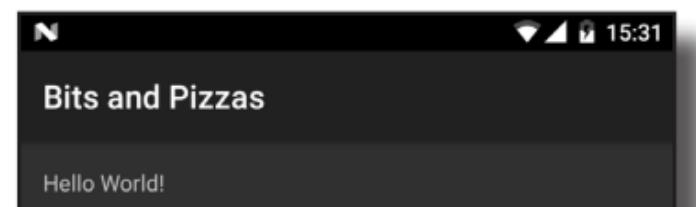
# APP Themes

---

Theme.MaterialComponents.Light



Theme.MaterialComponents



Theme.MaterialComponents.Light.NoActionBar



Theme.MaterialComponents.NoActionBar



# ActionBar vs. Toolbar

---

- Toolbar supports latest features
  - ActionBar (API 11+)
  - Toolbar (API 21+): Toolbar use material design theme features of Android and thus it provides backward compatibility up to API 7
- How to add toolbar
  1. Import androidx.appcompat.widget.Toolbar
  2. Remove existing app bar
  3. Add a toolbar to the layout
  4. Set the toolbar as your APP's new bar

# Components

- Here are some examples to help you use Components
  - <https://material.io/components>

The screenshot shows the 'Components' page of the Material Design website. The top navigation bar includes links for MATERIAL DESIGN, Design, Components (which is highlighted), Develop, Resources, Blog, and a search icon. A sidebar on the left lists various components: App bars: bottom, App bars: top, Backdrop, Banners, Bottom navigation, Buttons, Buttons: floating action button, Cards, Checkboxes, Chips, Data tables, and Date pickers. The main content area features a large title 'Components' and a sub-section titled 'Material Components are interactive building blocks for creating a user interface.' Below this, there's a link 'Browse all components or select a specific platform.' At the bottom of the main content, there are three cards illustrating 'App bars: bottom', 'App bars: top', and 'Backdrop'.

Components

Design Components Develop Resources Blog

Components

App bars: bottom

App bars: top

Backdrop

Banners

Bottom navigation

Buttons

Buttons: floating action button

Cards

Checkboxes

Chips

Data tables

Date pickers

All Components Android Web Flutter iOS

# Components

Material Components are interactive building blocks for creating a user interface.

Browse all components or select a specific platform.

**App bars: bottom**

A bottom app bar displays navigation and key actions at the bottom of mobile screens

**App bars: top**

The top app bar displays information and actions relating to the current screen

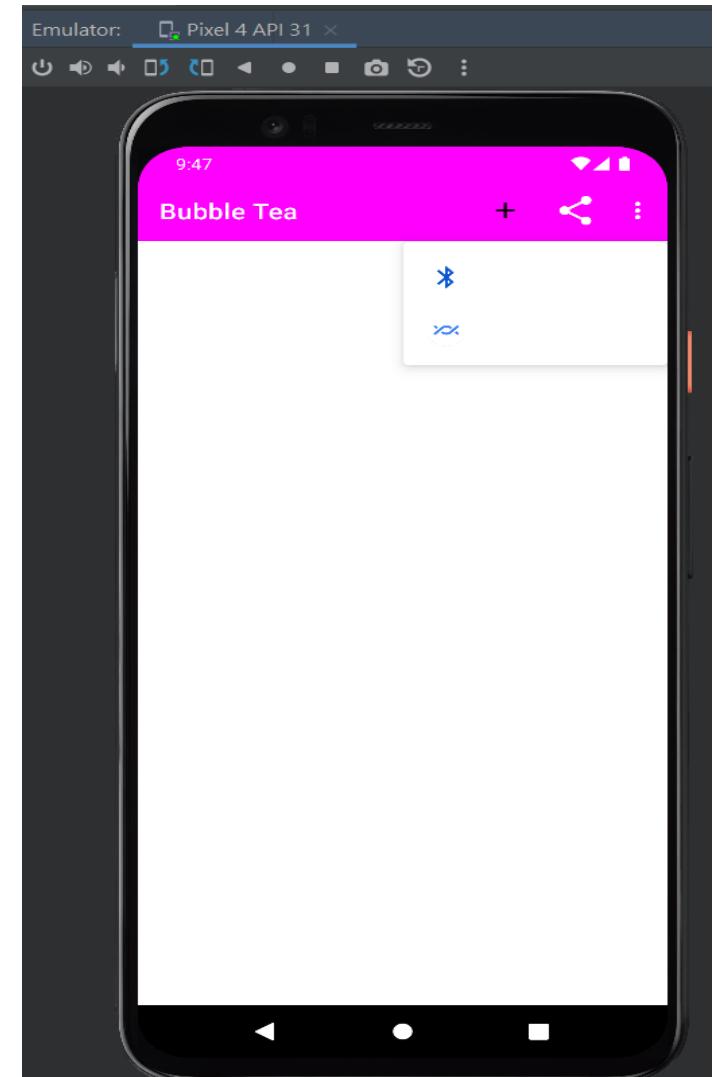
**Backdrop**

A backdrop appears behind all other surfaces in an app, displaying contextual and actionable content

# Today's Lab

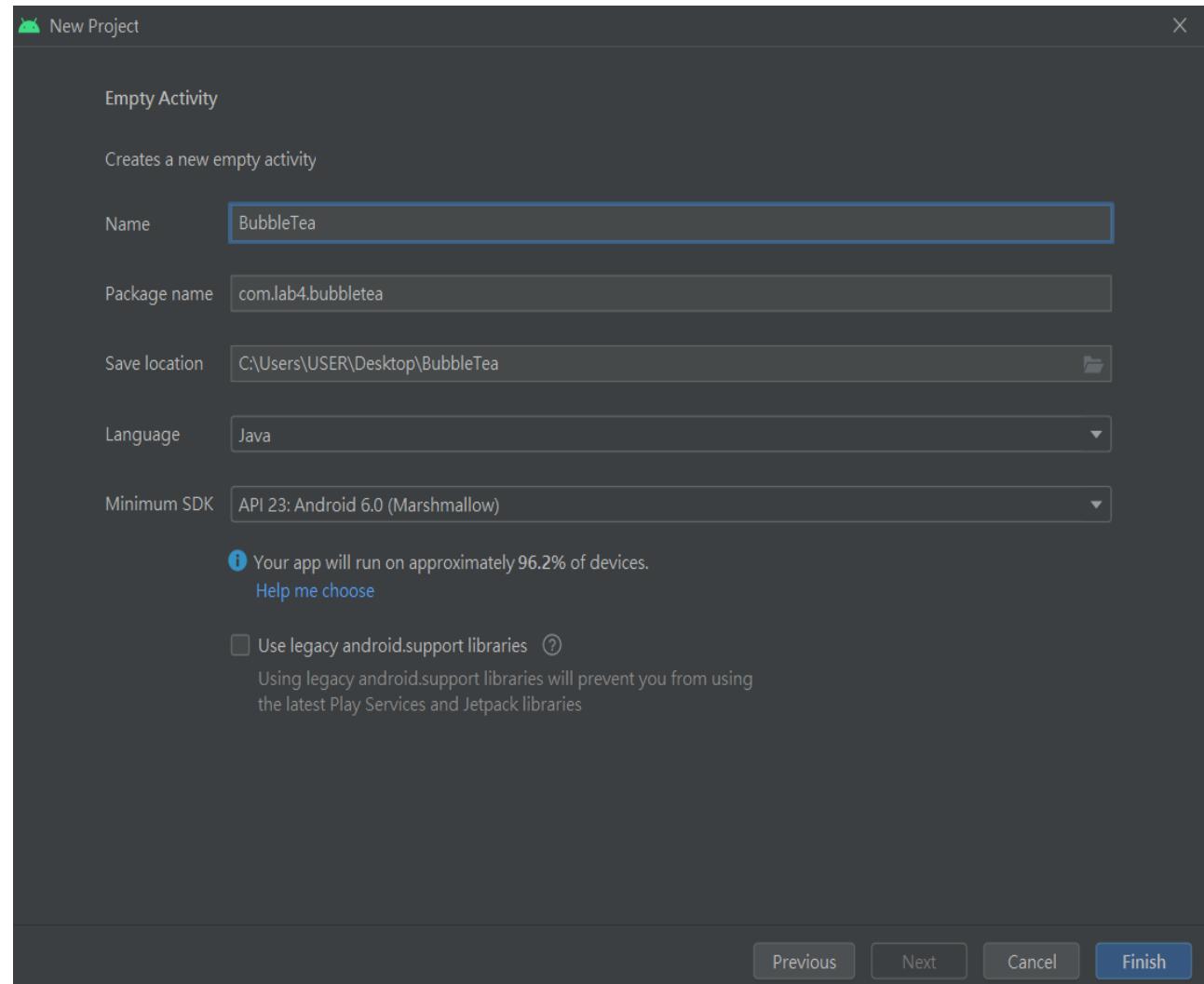
---

- A Bubble Tea APP that can create order and share messages
- Has special (terrible) color of APP bar



# Create BubbleTea Project

- Create a new project names **BubbleTea**
- Select **Empty Activity**
- Finish



# app/res/values/themes/themes.xml

- Remove action bar

styles.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

AndroidManifest.xml

```
<application
    ...
    android:theme="@style/AppTheme"
    ...
</application>
```

# app/res/values/color.xml

---

- Change colors

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#FF00FF</color>
    <color name="colorPrimaryDark">#FF00FF</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```



# Use Customized Toolbar

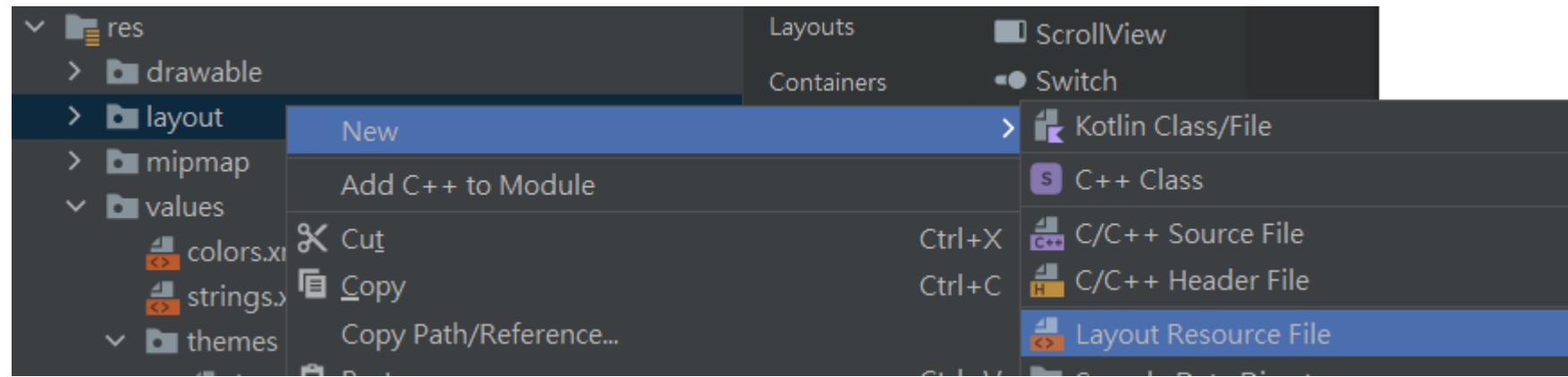
---

- Remove action bar
- Create app/res/layout/toolbar\_main.xml

# Create app/res/layout/toolbar\_main.xml

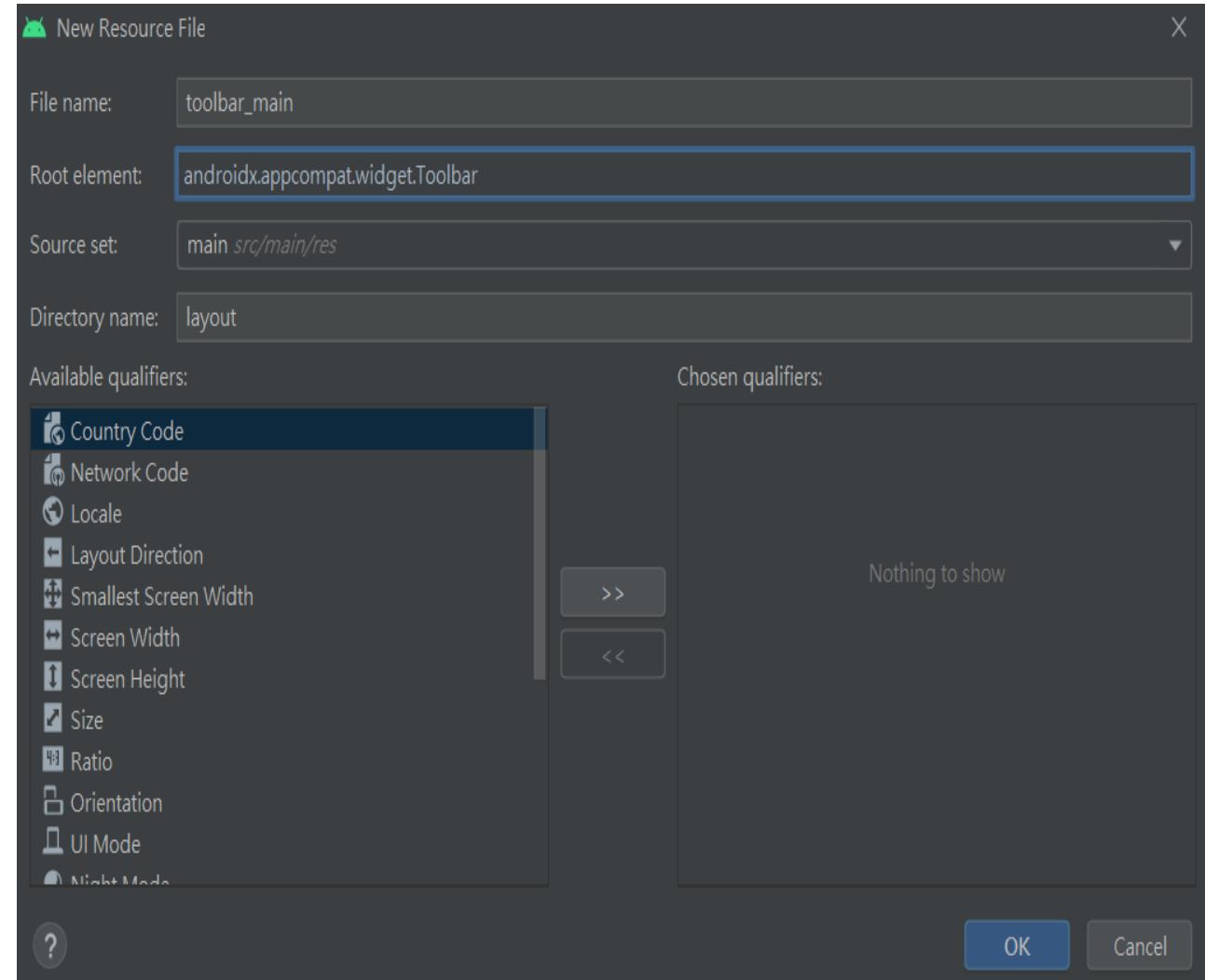
---

- Right Click layout folder -> New -> Layout resource file

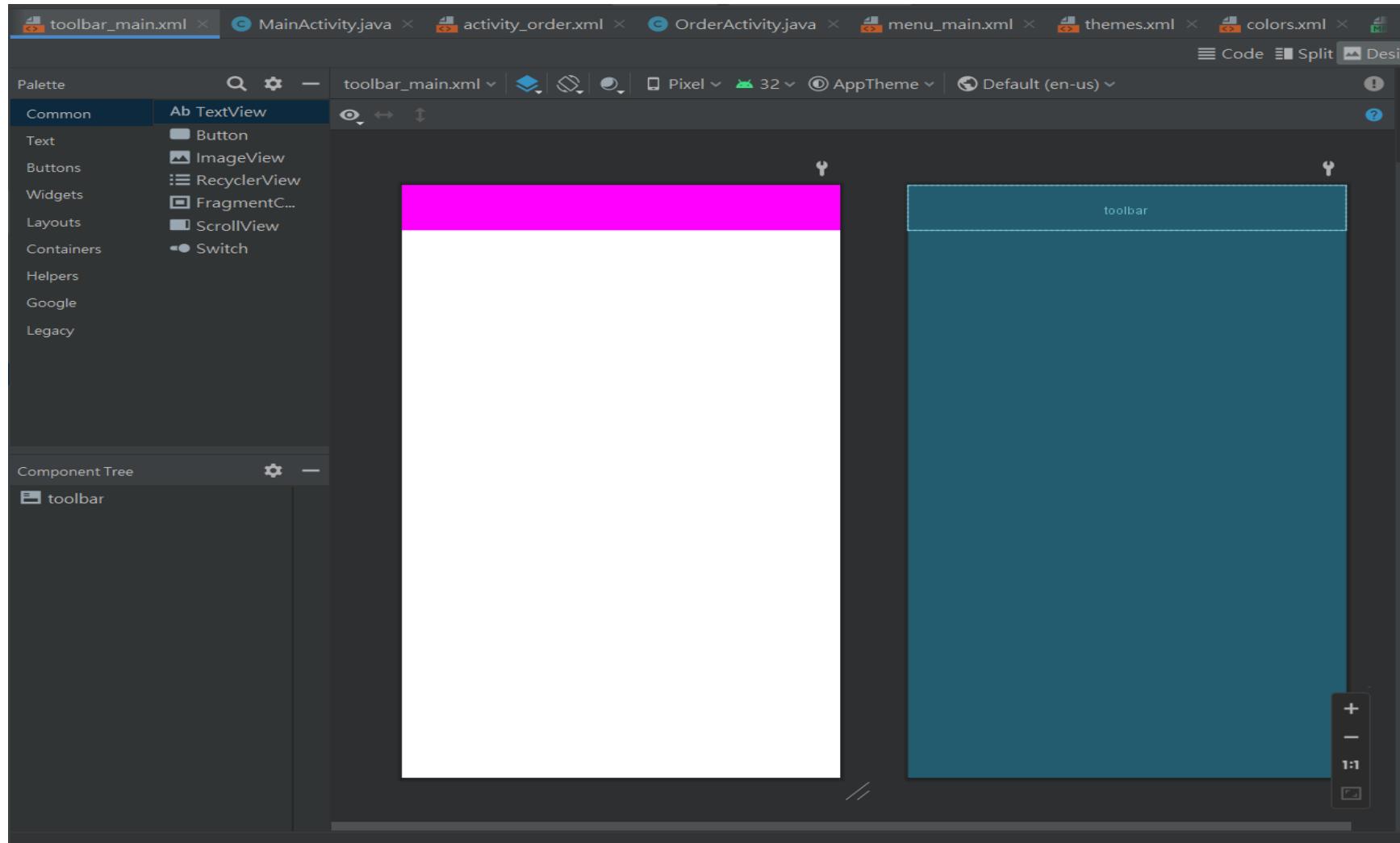


# Create app/res/layout/toolbar\_main.xml (Cont.)

- File name:
  - **toolbar\_main**
- Root element:
  - **androidx.appcompat.widget.Toolbar**



# Add Toolbar Widget



# toolbar\_main.xml

---

- Add toolbar widget in “toolbar\_main.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="?attr/colorPrimary"
    android:theme="@style/ThemeOverlay.MaterialComponents.Dark.ActionBar"
/>
```

# activity\_main.xml

---

- Change to LinearLayout
- Include our layout
  - <include layout="@layout/toolbar\_main">

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <include layout="@layout/toolbar_main" android:id="@+id/toolbar"/>
</LinearLayout>
```

# Add our toolbar in onCreate()

MainActivity.java

```
package com.lab4.bubbletea;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.os.Bundle;

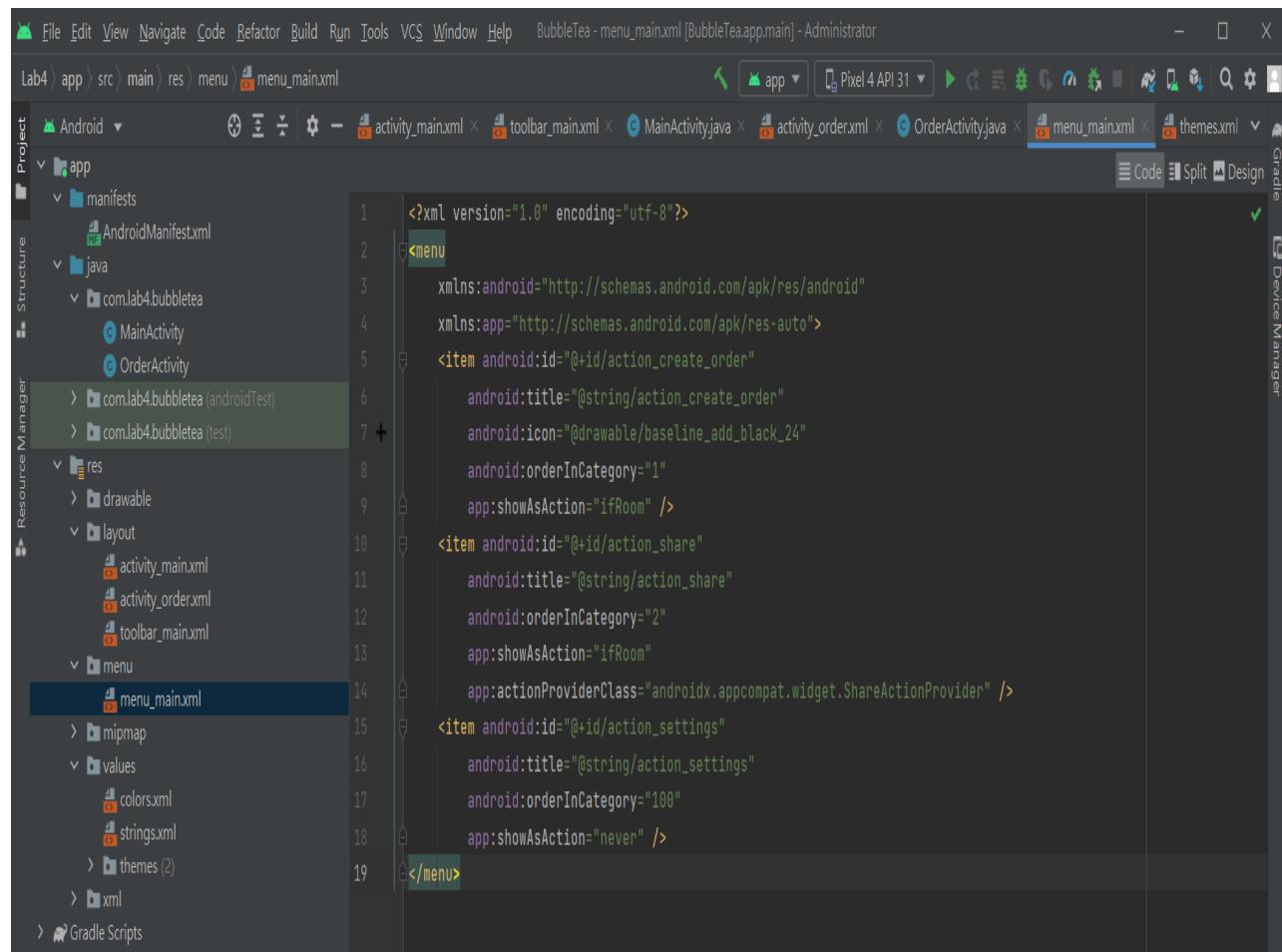
public class MainActivity extends AppCompatActivity {

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }
}
```

# Add Menu Buttons

- Create folder app/res/menu/
- Create file “menu\_main.xml”



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the "app" module. It includes "manifests", "java" (with "com.lab4.bubbletea" package containing "MainActivity" and "OrderActivity"), "res" (containing "drawable", "layout", "menu" (selected), "mipmap", "values", and "themes"), and "xml".
- Code Editor:** The main editor window displays the XML code for "menu\_main.xml".
- Code Content:**

```
<?xml version="1.0" encoding="utf-8"?>
<menu>
    <item android:id="@+id/action_create_order"
          android:title="@string/action_create_order"
          android:icon="@drawable/baseline_add_black_24"
          android:orderInCategory="1"
          app:showAsAction="ifRoom" />
    <item android:id="@+id/action_share"
          android:title="@string/action_share"
          android:orderInCategory="2"
          app:showAsAction="ifRoom"
          app:actionProviderClass="androidx.appcompat.widget.ShareActionProvider" />
    <item android:id="@+id/action_settings"
          android:title="@string/action_settings"
          android:orderInCategory="100"
          app:showAsAction="never" />
</menu>
```

Note :

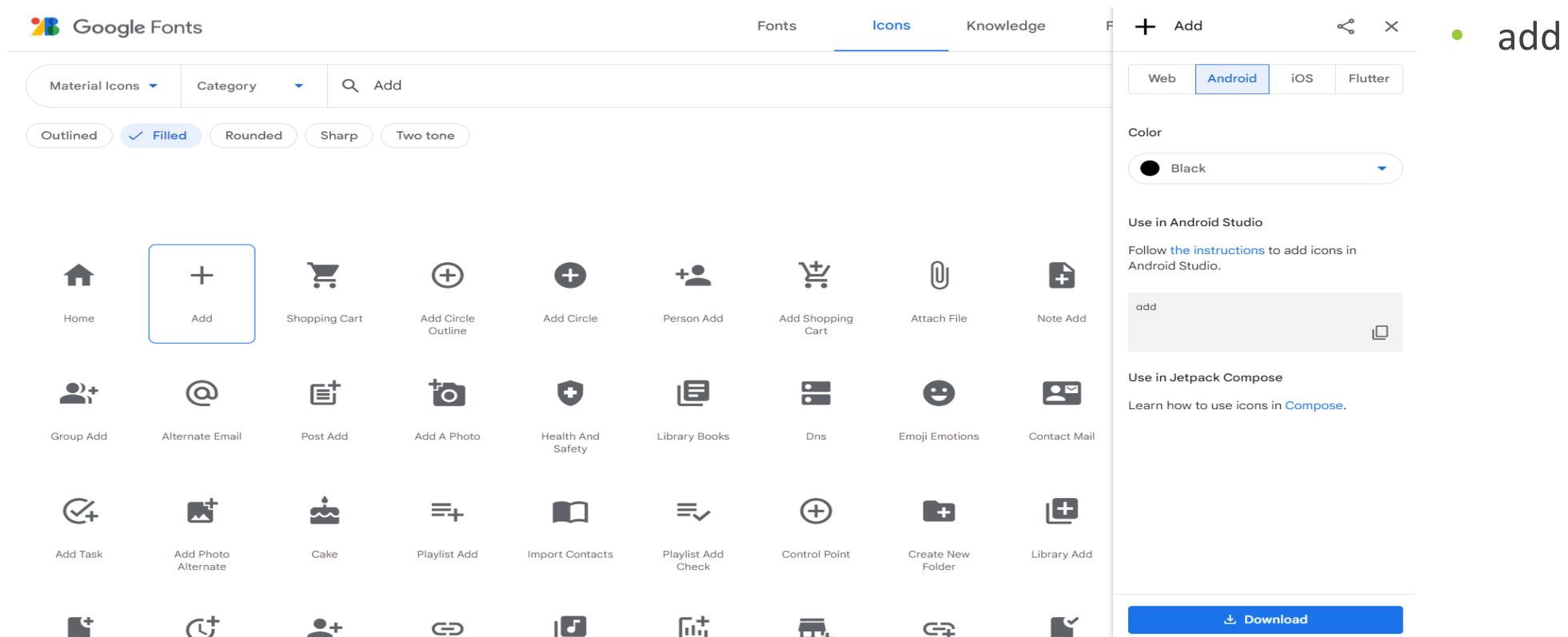
You won't see empty folders in Android Studio

# app/res/menu/menu\_main.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/action_create_order"
        android:title="@string/action_create_order"
        android:icon="@drawable/baseline_add_black_24"
        android:orderInCategory="1"
        app:showAsAction="ifRoom" />
    <item android:id="@+id/action_share"
        android:title="@string/action_share"
        android:orderInCategory="2"
        app:showAsAction="ifRoom"
        app:actionProviderClass=" androidx.appcompat.widget.ShareActionProvider" />
    <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100"
        app:showAsAction="never" />
</menu>
```

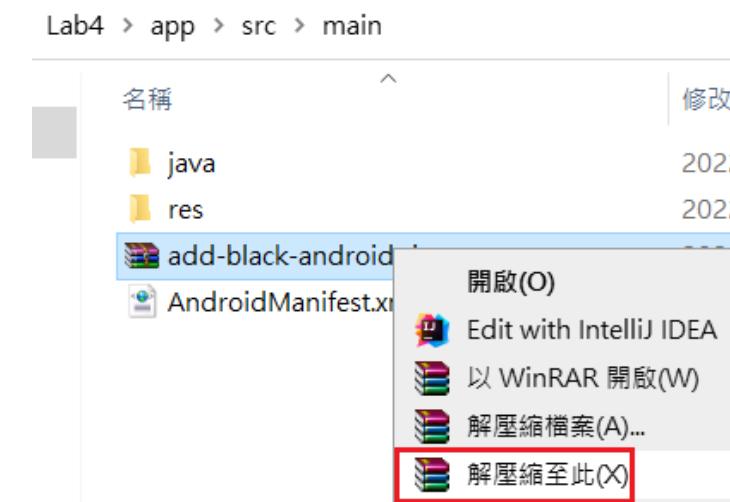
# Download icons

- <https://fonts.google.com/icons?selected=Material+Icons>
- Select your icon
- Select Android & Black at right side

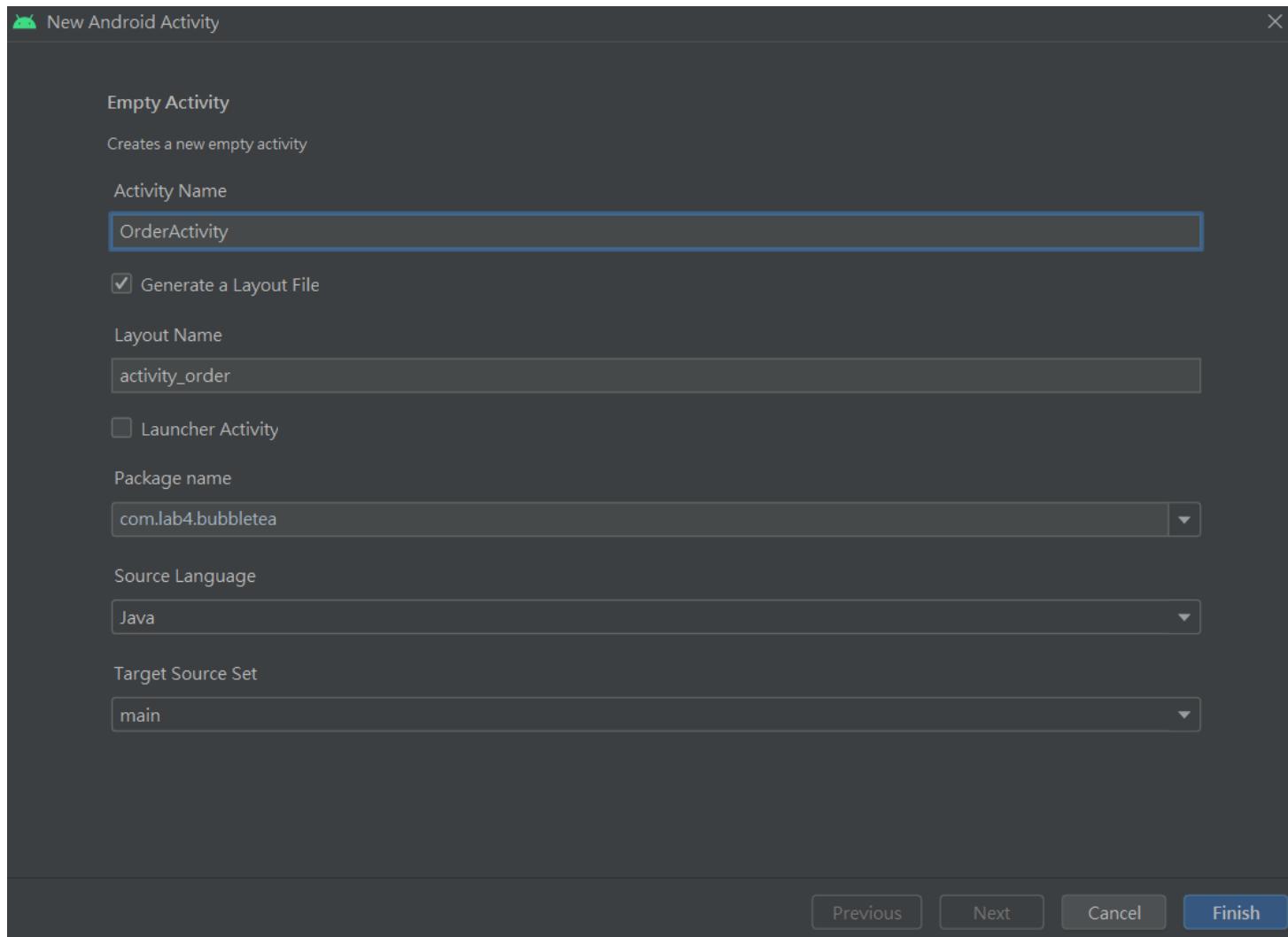


# Add icons to projects

- Copy all icon zip files to “./app/src/main”
- Decompress all zip files
- Images with different sizes and resolutions will be put in corresponding folders



# Create OrderActivity



# Include Toolbar in OrderActivity

---

- Include toolbar in app/res/layout/activity\_order.xml

activity\_order.xml

```
<androidx.constraintlayout.widget.ConstraintLayout ...>
    <include layout="@layout/toolbar_main" android:id="@+id/toolbar"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Update AndroidManifest.xml

```
<activity
    android:name=".OrderActivity"
    android:exported="false"
    android:parentActivityName=".MainActivity" />
```

# Add our toolbar in onCreate() of OrderActivity

OrderActivity.java

```
public class OrderActivity extends AppCompatActivity {

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar actionBar = getSupportActionBar();
        actionBar.setDisplayHomeAsUpEnabled(true);
    }
}
```

# onCreateOptionsMenu(Menu menu)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    ...  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.menu_main, menu);  
        return super.onCreateOptionsMenu(menu);  
    }  
}
```

# onOptionsItemSelected(MenuItem item)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    ...  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        switch (item.getItemId()) {  
            case R.id.action_create_order:  
                //Code to run when the Create Order item is clicked  
                Intent intent = new Intent(this, OrderActivity.class);  
                startActivity(intent);  
                return true;  
            case R.id.action_settings:  
                //Code to run when the settings item is clicked  
                return true;  
            default:  
                return super.onOptionsItemSelected(item);  
        }  
    }  
}
```

# Add Up(Back) Button



# Add Share Button

---

This is what the share action looks like on the app bar. When you click on it, it gives you a list of apps that you can use to share content.



The share action also displays an icon for the app you most commonly share content with, in this case the Messenger app. This may not be visible at first.

# app/res/values/strings.xml

---

```
<resources>
    <string name="app_name">Bubble Tea</string>
    <string name="action_settings">Settings</string>
    <string name="action_create_order">Create Order</string>
    <string name="title_activity_order">Order Activity</string>
    <string name="action_share">Share</string>
</resources>
```

# Add shareActionProvider in MainActivity

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private ShareActionProvider shareActionProvider;
    ...
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        ...
        MenuItem menuItem = menu.findItem(R.id.action_share);
        shareActionProvider = ( androidx.appcompat.widget.ShareActionProvider) MenuItemCompat.getActionProvider(menuItem);
        setShareActionIntent("Want to buy a bubble tea?");

        return super.onCreateOptionsMenu(menu);
    }

    private void setShareActionIntent(String text) {
        Intent intent = new Intent(Intent.ACTION_SEND);
        intent.setType("text/plain");
        intent.putExtra(Intent.EXTRA_TEXT, text);
        shareActionProvider.setShareIntent(intent);
    }
    ...
}
```