



Dialog - RecyclerView

342 267 Mobile Device Programming

Asst. Prof. Monlica Wattana, Ph.D
Department of Computer Science,
Khon Kaen University



Outline

- Dialog
- RecyclerView



Dialogs

- A dialog is a small window that prompts the user to **make a decision** or **enter additional information**.
- A dialog **does not fill the screen** and is normally used for modal events that require users to take an action before they can proceed.
- A dialog is always created and displayed as a part of an **Activity**.



Alert Dialog

- An **AlertDialog** is an extension of the Dialog class that contains:
 - Title Text
 - Message Text
 - Buttons – There are three types of them: Positive, Negative, Neutral
 - A list of selectable items (with optional checkboxes or radio buttons)



Alert Dialog

- To use AlertDialog in Android Studio project import the following:

```
import androidx.appcompat.app.AlertDialog
```

- To create an AlertDialog we use the **AlertDialog.Builder** inner class.

```
val alertDialogBuilder = AlertDialog.Builder(this)
```



Alert Dialog

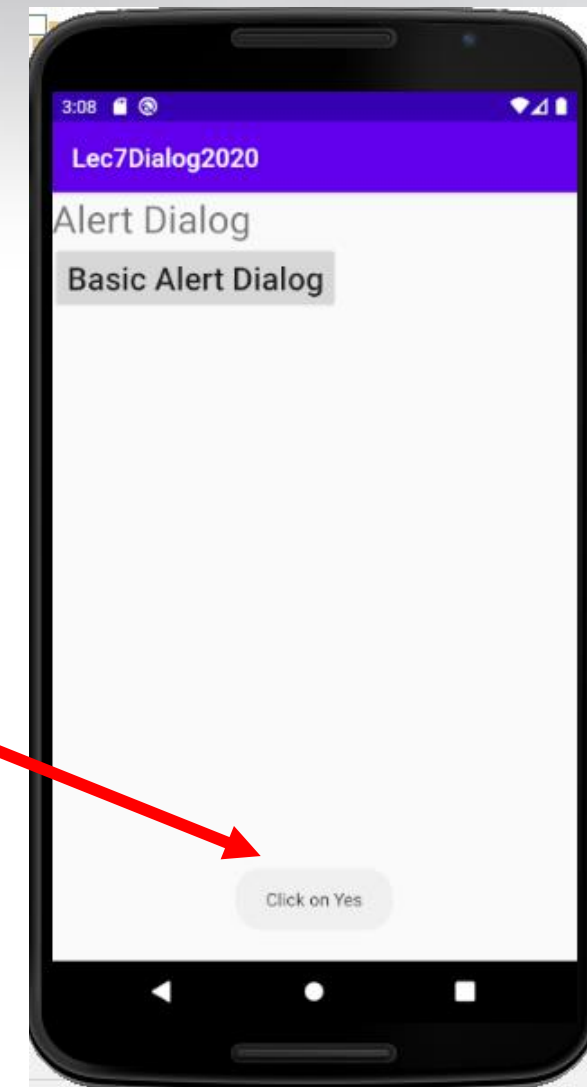
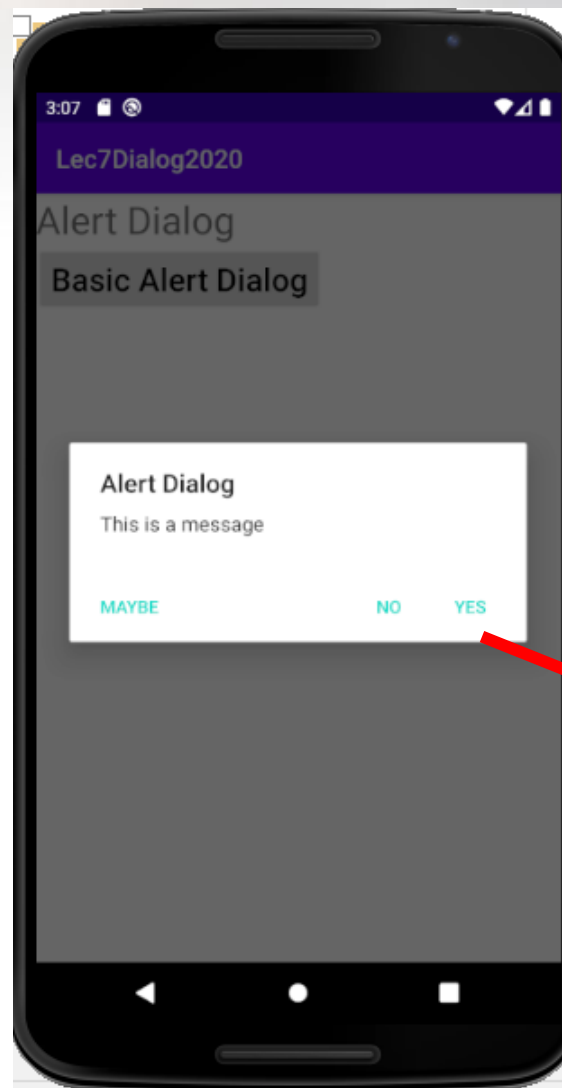
- The methods can be used on an AlertDialog.
 - **setTitle**: adds Title to the dialog
 - **setIcon**: adds icon to the dialog
 - **setMessage**: adds a message for the dialog
 - **setNegativeButton**: set a listener to be invoked when the negative button of the dialog is pressed.
 - **setPositiveButton**: Set a listener to be invoked when the positive button of the dialog is pressed.
 - **setNeutralButton**: set a listener to be invoked when the neutral button of the dialog is pressed.



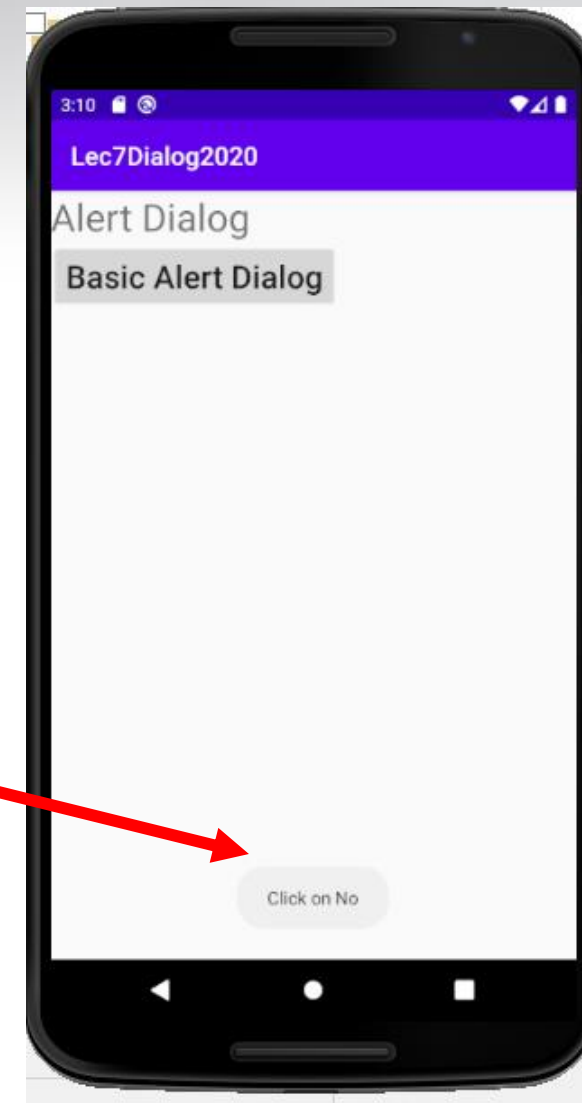
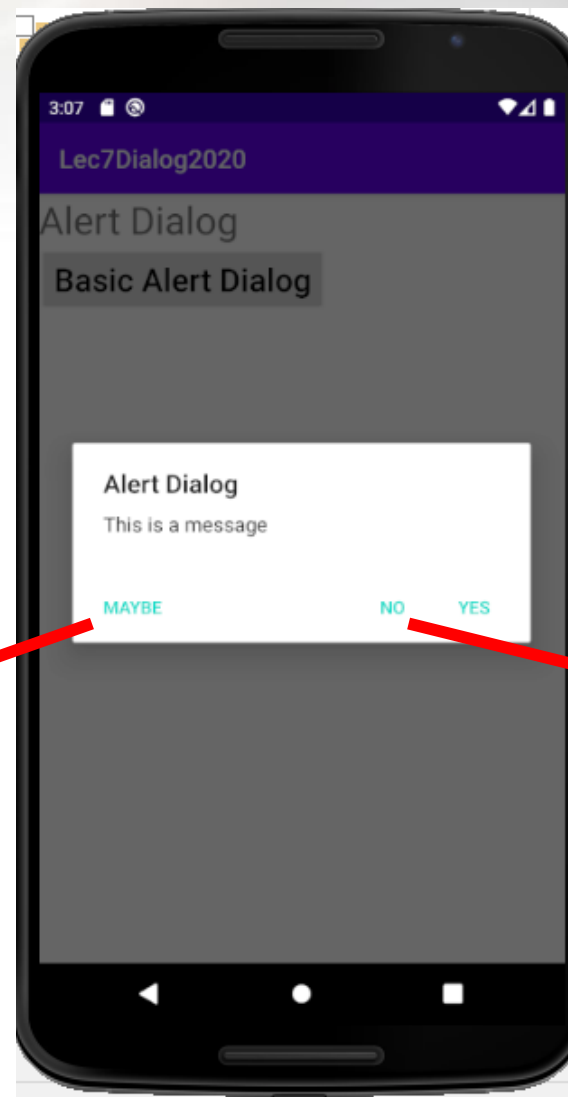
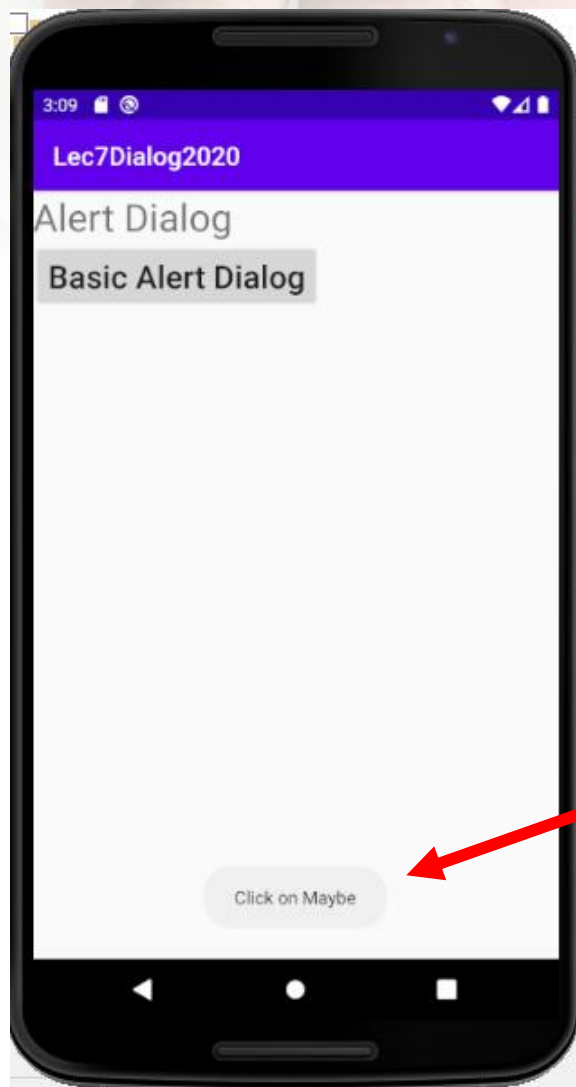
Alert Dialog

- The methods can be used on an AlertDialog.
 - `show()` – used to display the AlertDialog
 - `setList` – used to set an array of strings which would be displayed in the form of a List.
 - `setMultiChoiceList` – set an array to select multiple items from the List to CheckBox.

Basic Alert Dialog



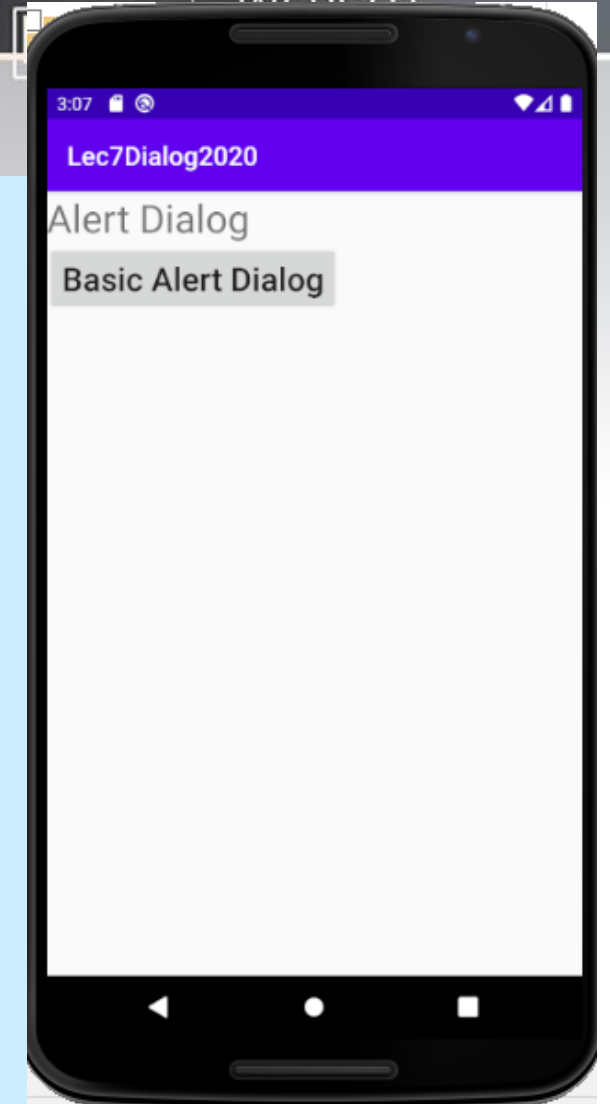
Basic Alert Dialog



activity_main.xml

Basic Alert Dialog

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alert Dialog"
        android:textSize="30sp" />
    <Button
        android:id="@+id/btnBasicAlert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Basic Alert Dialog"
        android:textAllCaps="false"
        android:textSize="25sp"/>
</LinearLayout>
```



Basic Alert Dialog

- MainActivity.kt

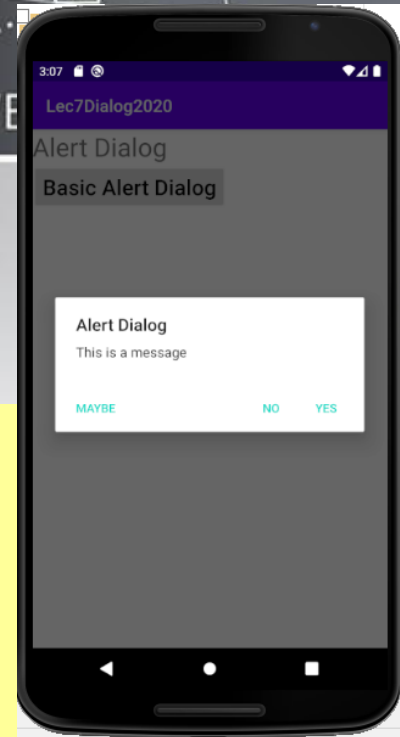
```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val btn_ClickAlert = findViewById<Button>(R.id.btnBasicAlert)  
        btn_ClickAlert.setOnClickListener{  
            showBasicAlert()  
        }  
    }  
}
```



Basic Alert Dialog

- MainActivity.kt(cont.)

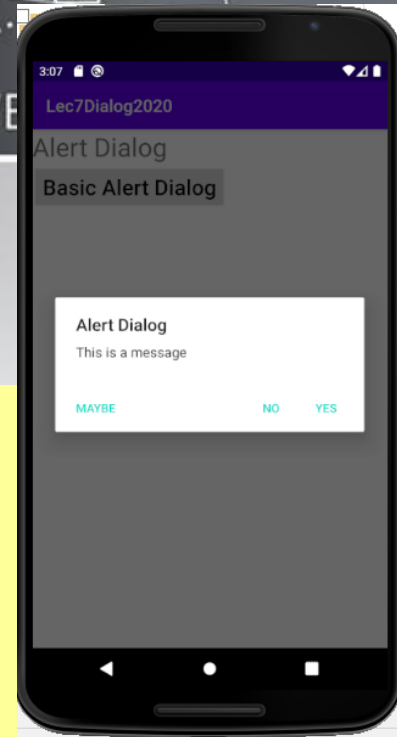
```
fun showBasicAlert(){  
    val myBuilder = AlertDialog.Builder(this)  
        myBuilder.setTitle("Alert Dialog")  
        myBuilder.setMessage("This is a message")  
        myBuilder.setPositiveButton("Yes") { dialog, which ->  
            Toast.makeText(applicationContext, "Click on Yes", Toast.LENGTH_SHORT).show()  
        }  
        myBuilder.setNegativeButton("No"){_, _ ->  
            Toast.makeText(applicationContext, "Click on No", Toast.LENGTH_SHORT).show()  
        }  
        myBuilder.setNeutralButton("Maybe"){_, _ ->  
            Toast.makeText(applicationContext, "Click on Maybe", Toast.LENGTH_SHORT).show()  
        }  
    myBuilder.show()  
}
```



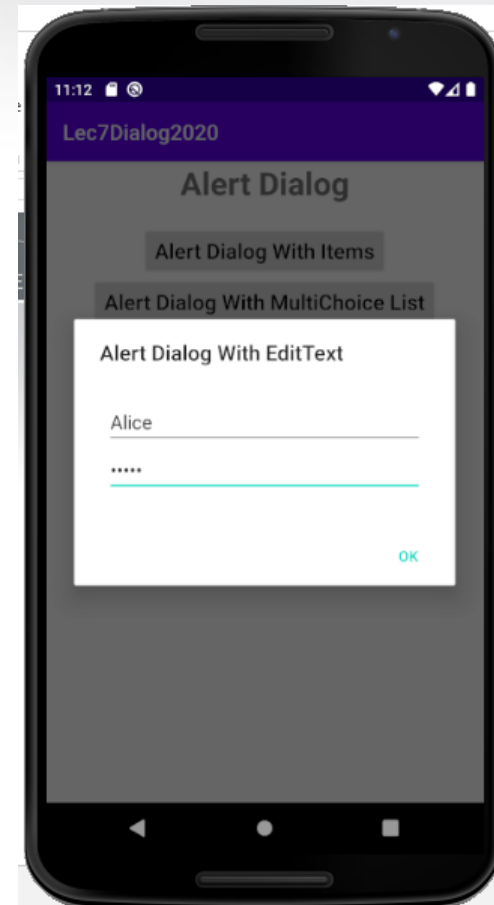
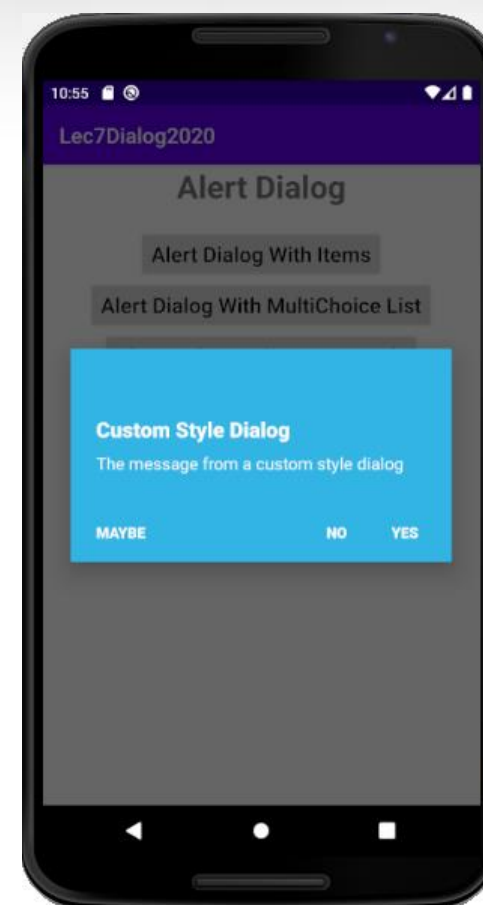
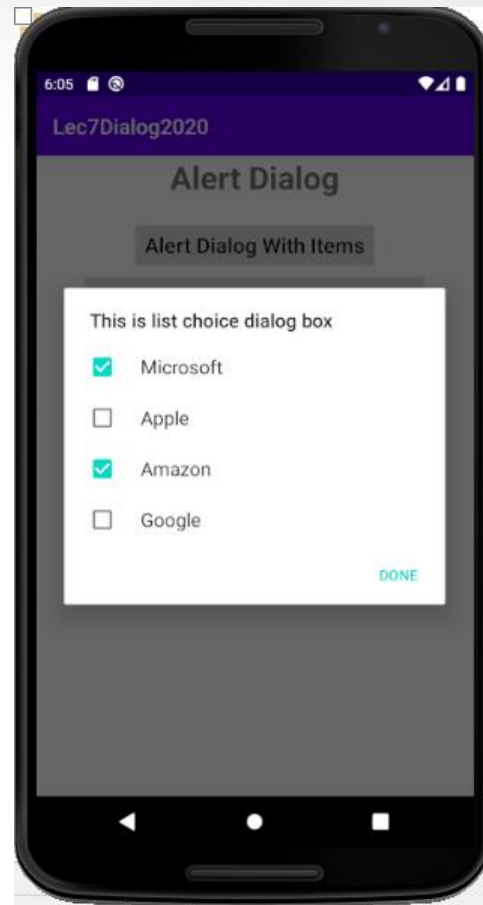
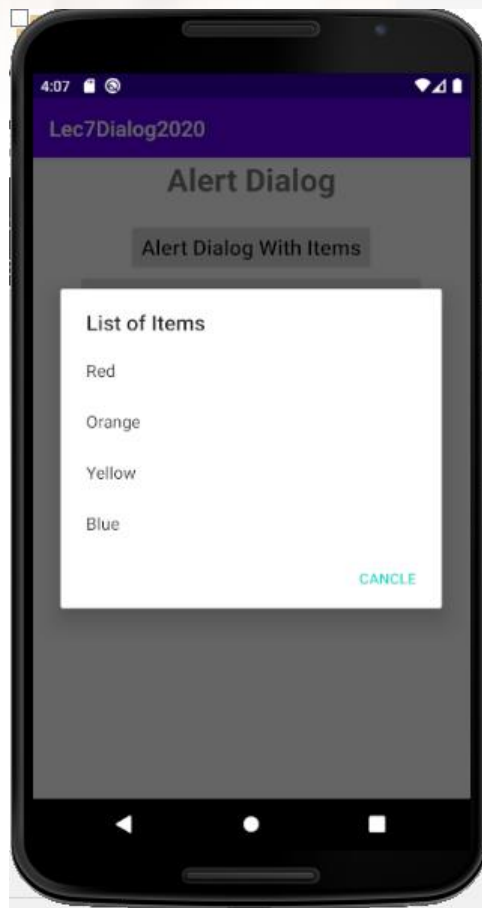
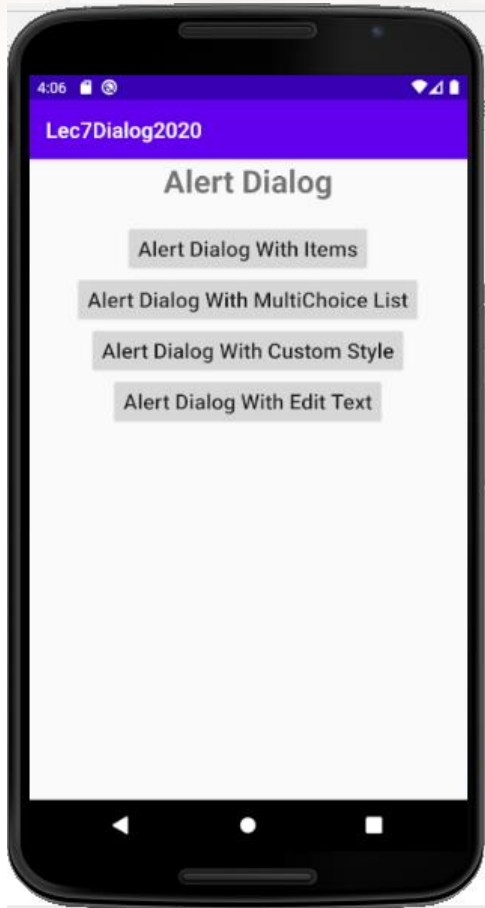
Basic Alert Dialog

- MainActivity.kt(cont. **apply** Method Version)

```
fun showBasicAlert(){ // Use apply
    val myBuilder = AlertDialog.Builder(this)
    myBuilder.apply {
        setTitle("Alert Dialog")
        setMessage("This is a message")
        setPositiveButton("Yes") { dialog, which ->
            Toast.makeText(applicationContext, "Click on Yes", Toast.LENGTH_SHORT).show()
        }
        setNegativeButton("No"){_, _ ->
            Toast.makeText(applicationContext, "Click on No", Toast.LENGTH_SHORT).show()
        }
        setNeutralButton("Maybe"){_, _ ->
            Toast.makeText(applicationContext, "Click on Maybe", Toast.LENGTH_SHORT).show()
        }
        show()
    }
}
```



Alert Dialog Example: activity_main.xml



Alert Dialog Example: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout .....
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:gravity="center_horizontal"
```

```
    android:orientation="vertical"
```

```
    tools:context=".MainActivity">
```

```
<TextView
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Alert Dialog"
```

```
    android:textStyle="bold"
```

```
    android:textSize="30sp"
```

```
    android:textAlignment="center"/>
```

```
<Button
```

```
    android:id="@+id/btnAlertWithItems"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

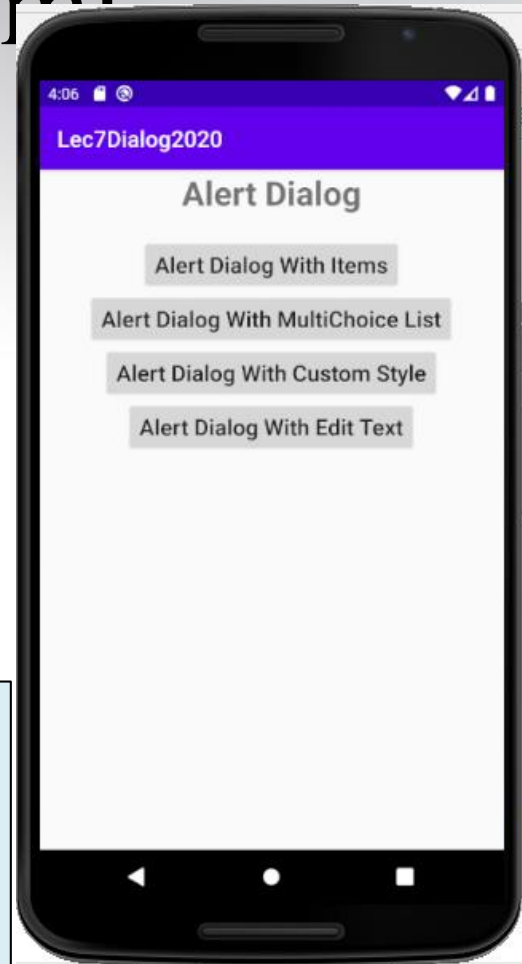
```
    android:onClick="withItems"
```

```
    android:text="Alert Dialog With Items"
```

```
    android:textAllCaps="false"
```

```
    android:layout_marginTop="20dp"
```

```
    android:textSize="20sp"/>
```

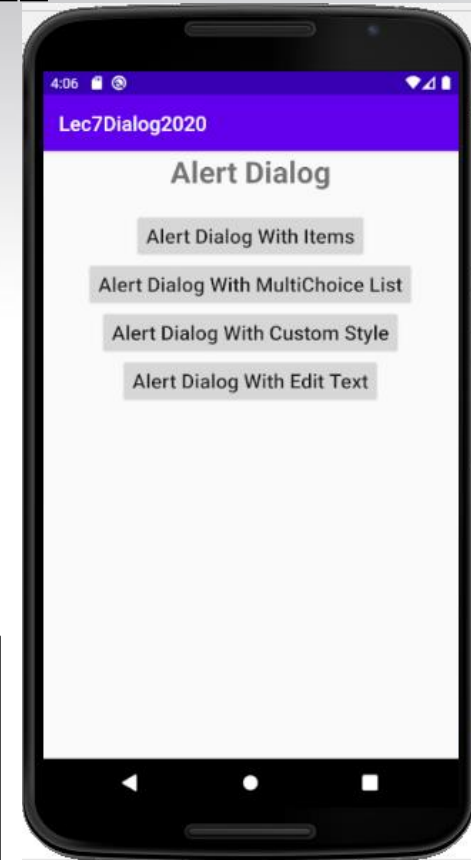


Alert Dialog Example : activity_main.xml

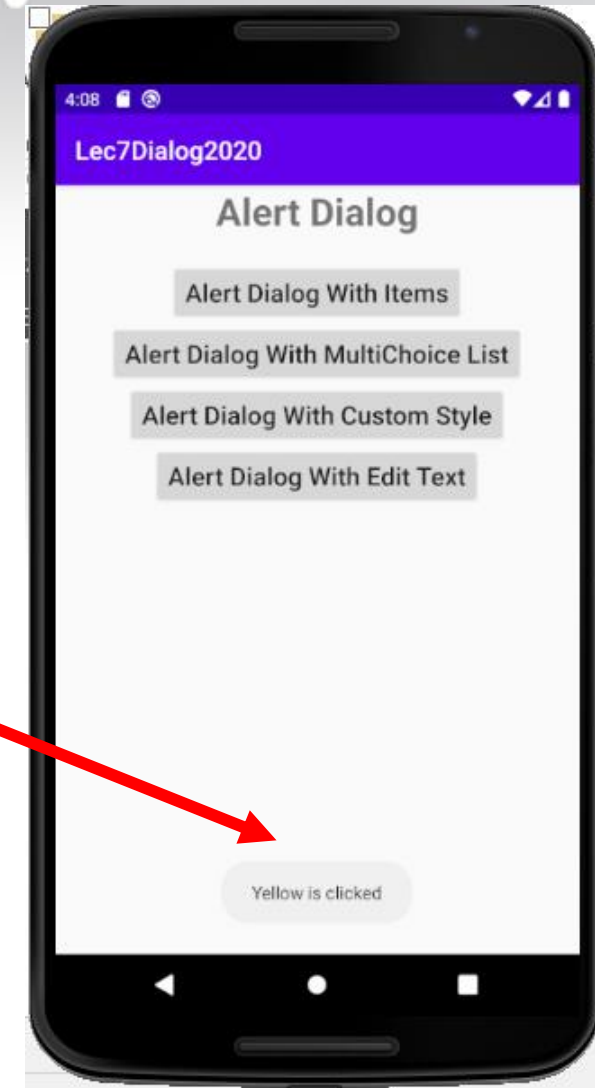
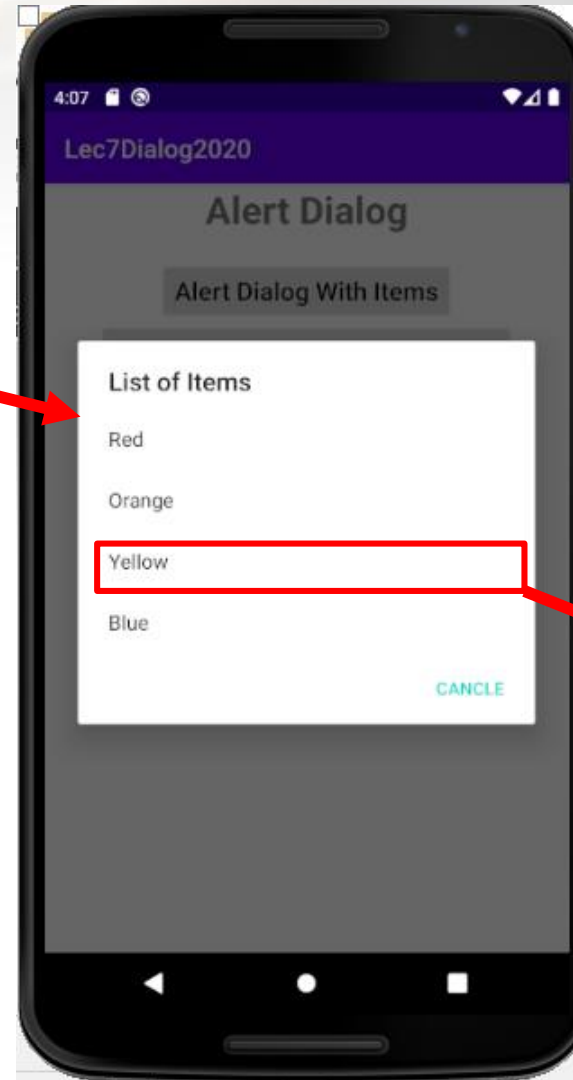
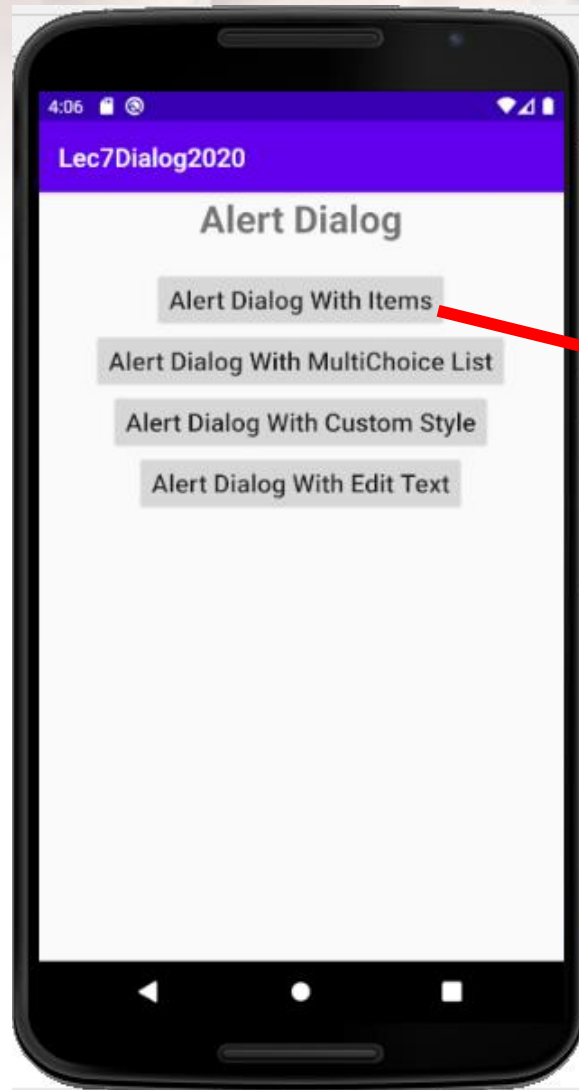
```
<Button
    android:id="@+id/btnAlertWithMultiChoiceList"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="withMultiChoiceList"
    android:text="Alert Dialog With MultiChoice List"
    android:textAllCaps="false"
    android:textSize="20sp"/>
```

```
<Button
    android:id="@+id/btnAlertWithCustomStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="withCustomStyle"
    android:text="Alert Dialog With Custom Style"
    android:textAllCaps="false"
    android:textSize="20sp"/>
```

```
<Button
    android:id="@+id/btnAlertWithEditText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="withEditText"
    android:text="Alert Dialog With Edit Text"
    android:textAllCaps="false"
    android:textSize="20sp"/>
</LinearLayout>
```

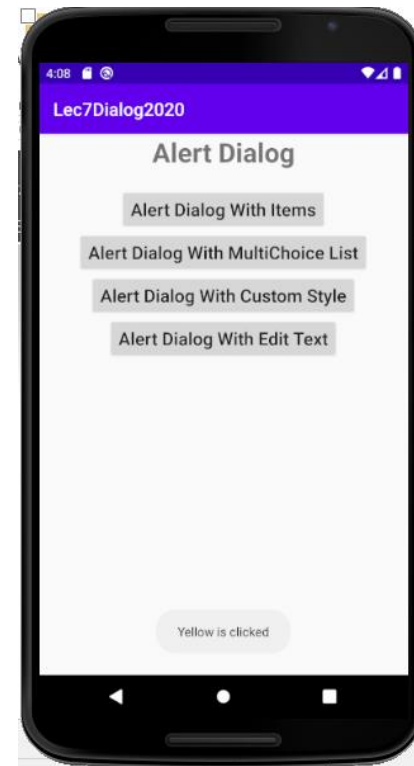
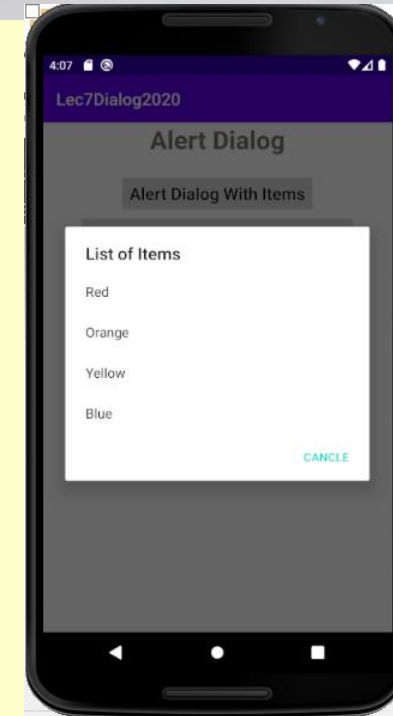


Alert Dialog With Items

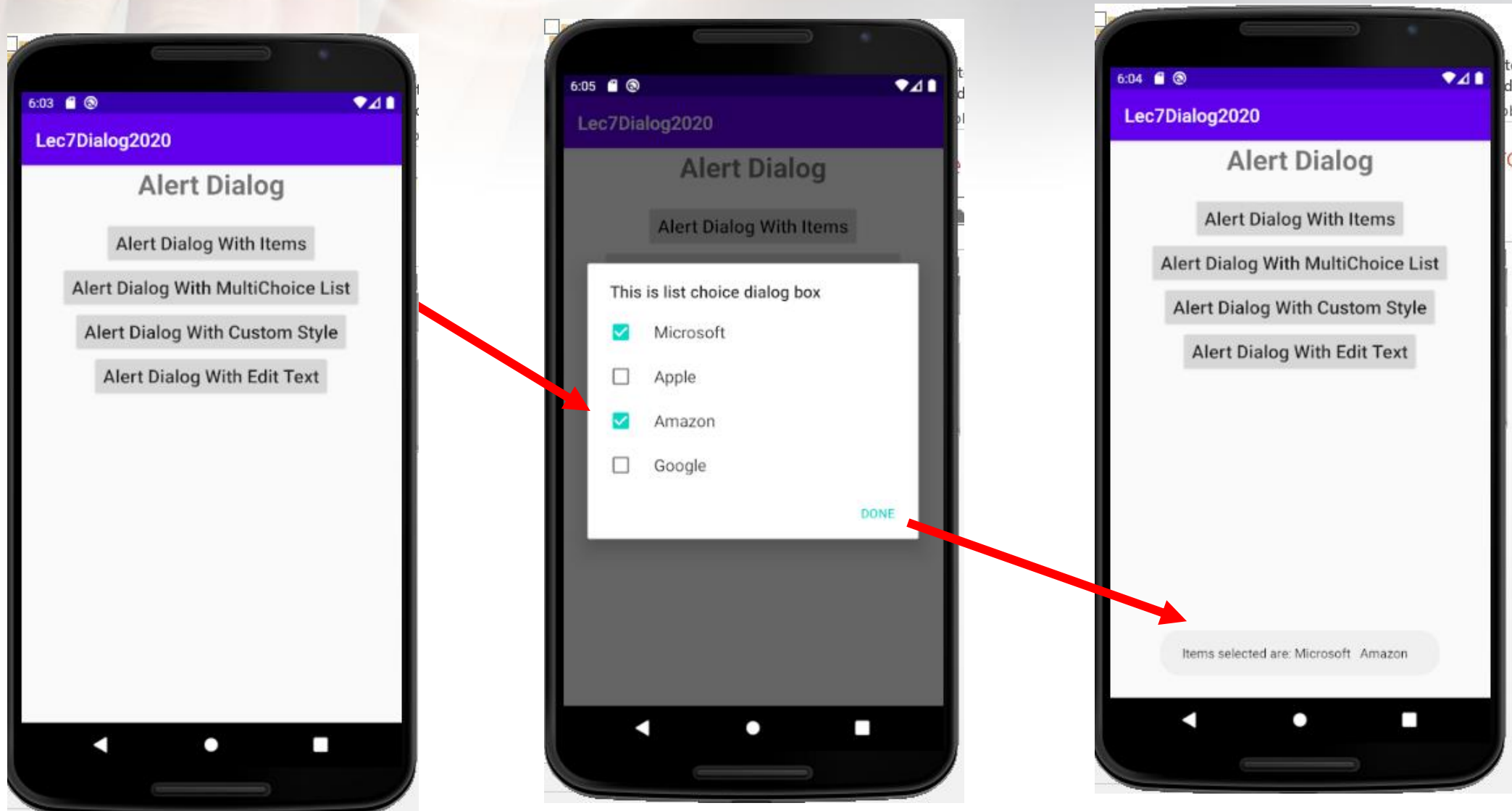


Alert Dialog With Items: MainActivity.kt

```
fun withItems(v: View){  
    val items = arrayOf("Red", "Orange", "Yellow", "Blue")  
    val myBuilder = AlertDialog.Builder(this)  
    myBuilder.apply {  
        setTitle("List of Items")  
        setItems(items) { dialog, which ->  
            Toast.makeText(applicationContext, items[which] + " is clicked",  
                Toast.LENGTH_SHORT).show()  
        }  
        setPositiveButton("Cancle"){ _, _ ->  
            Toast.makeText(applicationContext,"Click on Cancle", Toast.LENGTH_SHORT).show()  
        }  
        show()  
    }  
}
```

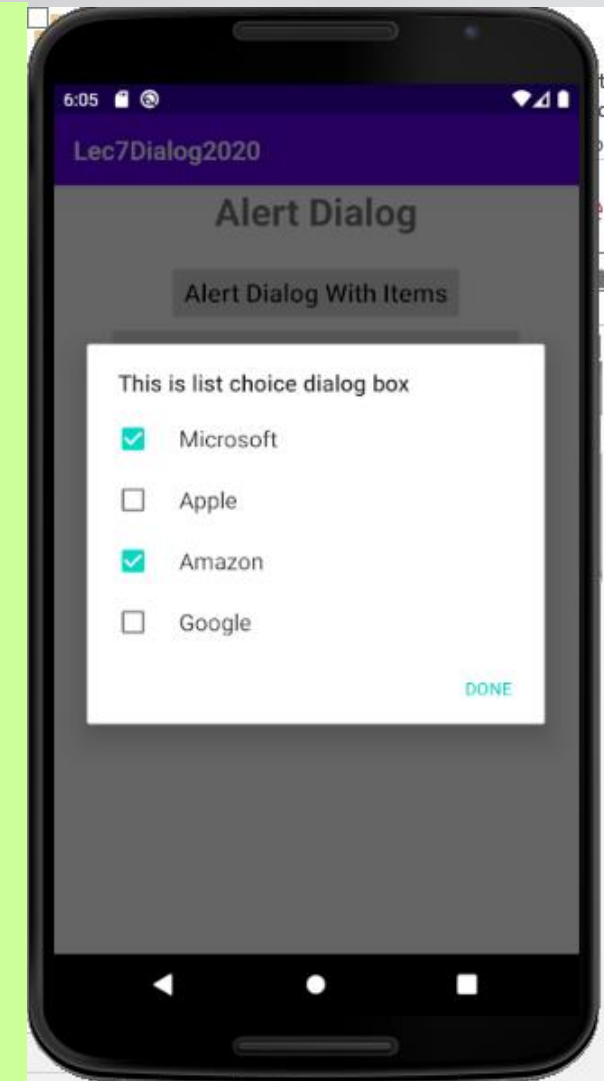


Alert Dialog With MultiChoice List



Alert Dialog With MultiChoice List: MainActivity.kt

```
fun withMultiChoiceList(v: View){  
    val items = arrayOf("Microsoft", "Apple", "Amazon", "Google")  
    val selectedList = ArrayList<Int>() // collect selected item  
    var str :String ="" // show selected item  
    val myBuilder = AlertDialog.Builder(this)  
    myBuilder.apply {  
        setTitle("This is list choice dialog box")  
        setMultiChoiceItems(items, null) { dialog, which, isChecked ->  
            if (isChecked) {  
                selectedList.add(which)  
            } else if (selectedList.contains(which)) {  
                selectedList.remove(Integer.valueOf(which))  
            }  
        }  
    }  
}
```

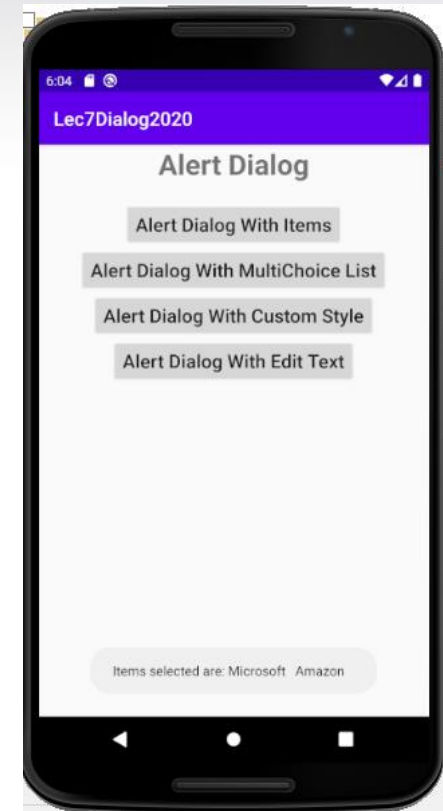
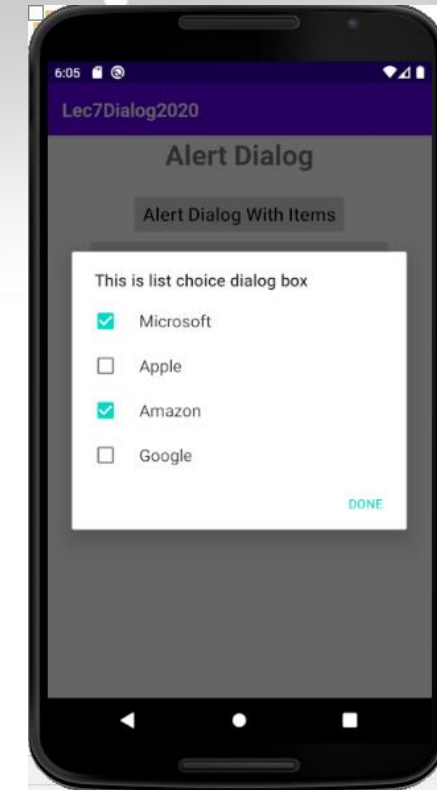


Alert Dialog With MultiChoice List: MainActivity.kt

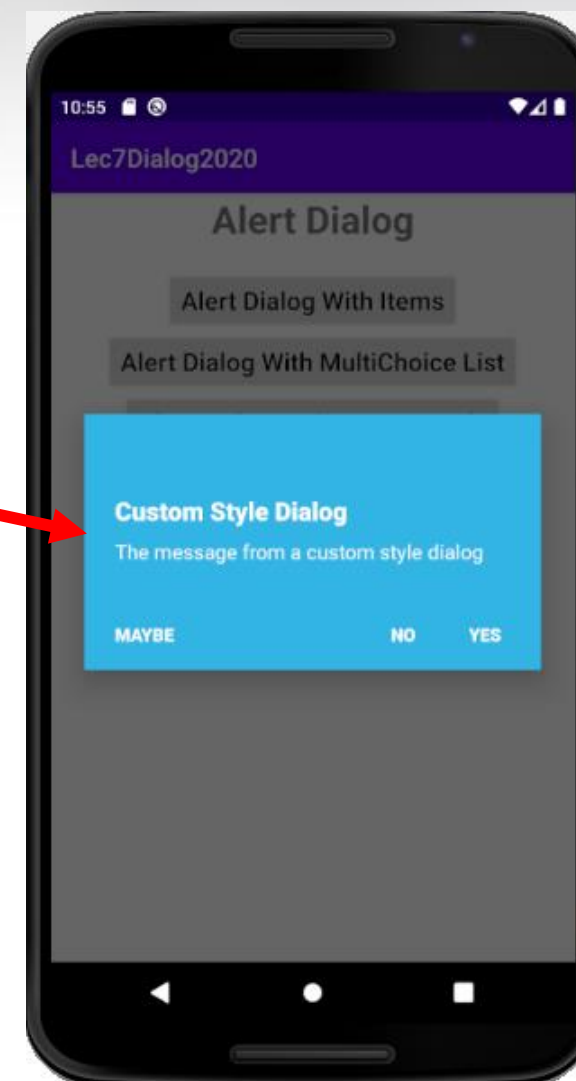
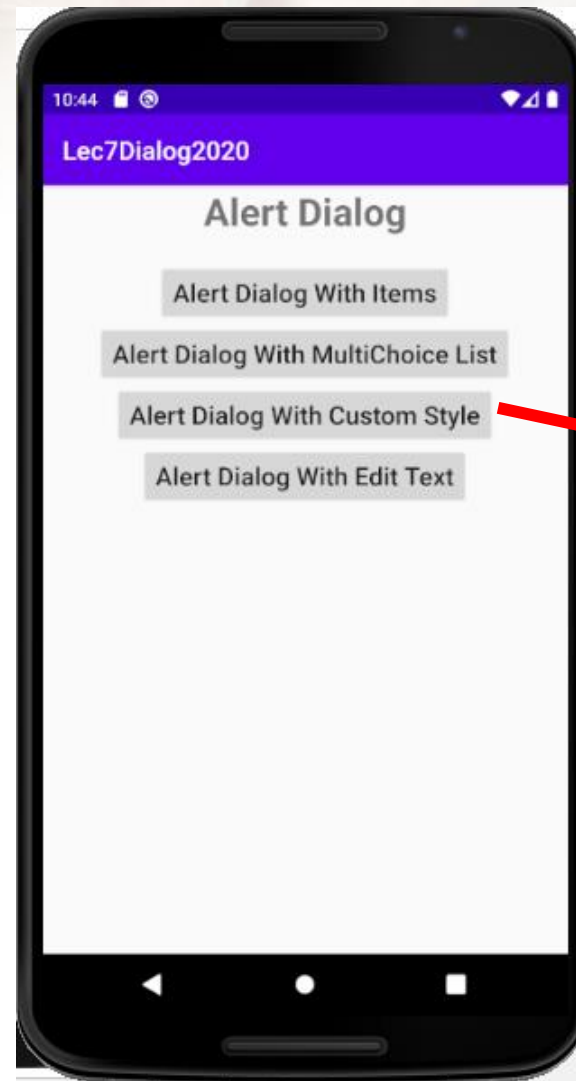
```
setPositiveButton("DONE") { dialogInterface, i ->

    for (j in selectedList.indices) {
        str += items[selectedList[j]].toString() + " "
    }
    Toast.makeText(applicationContext, "Items selected are: $str",
        Toast.LENGTH_SHORT).show()

}
show()
}
```

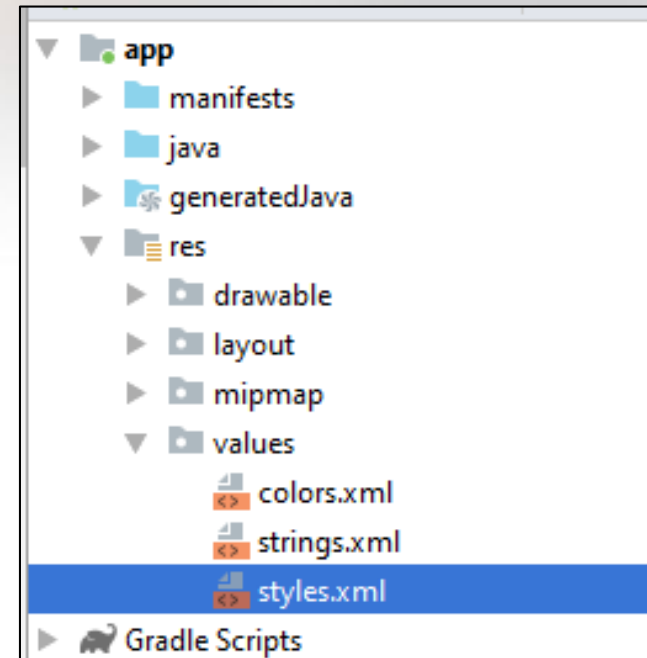


Alert Dialog With Custom Style



Alert Dialog With Custom Style: styles.xml

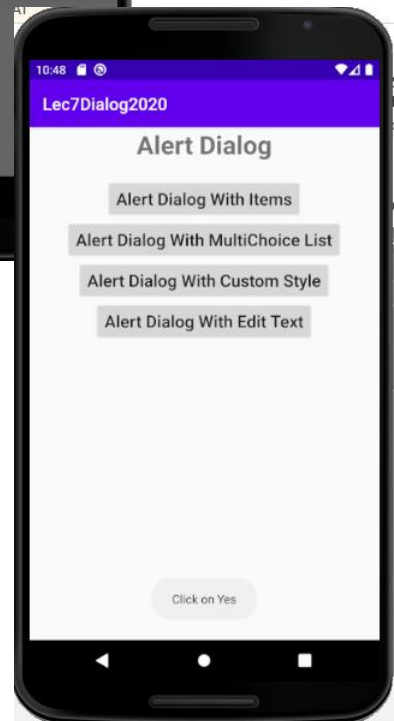
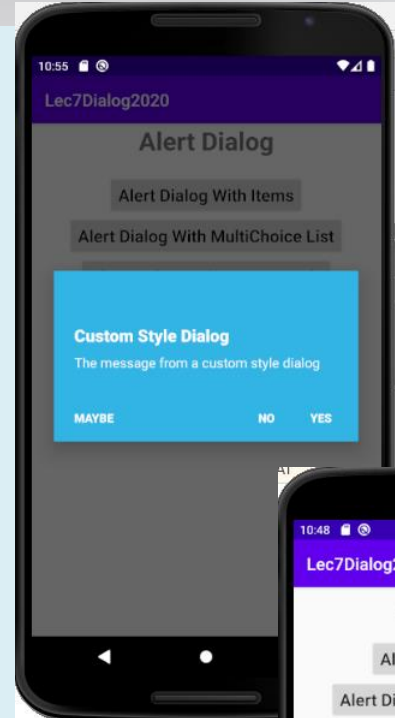
Res >> value >> styles.xml



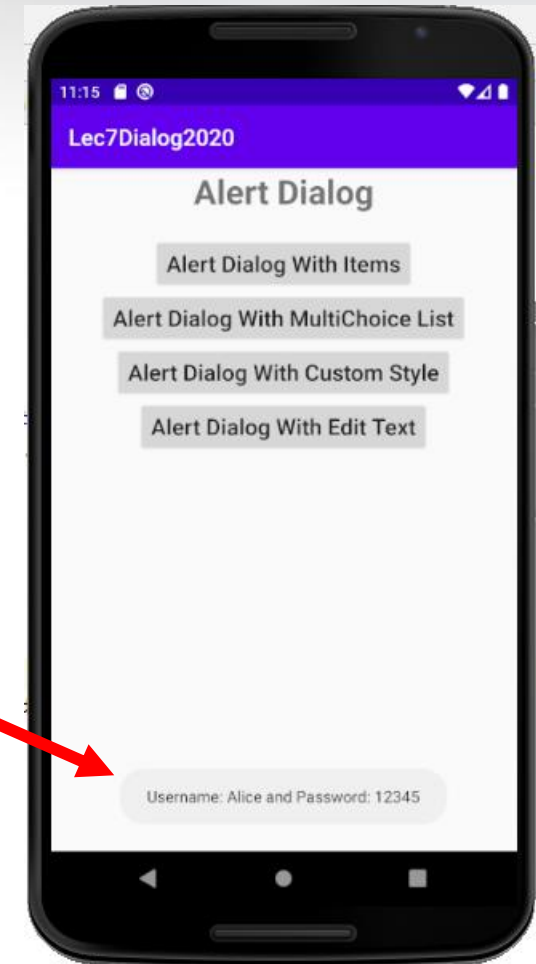
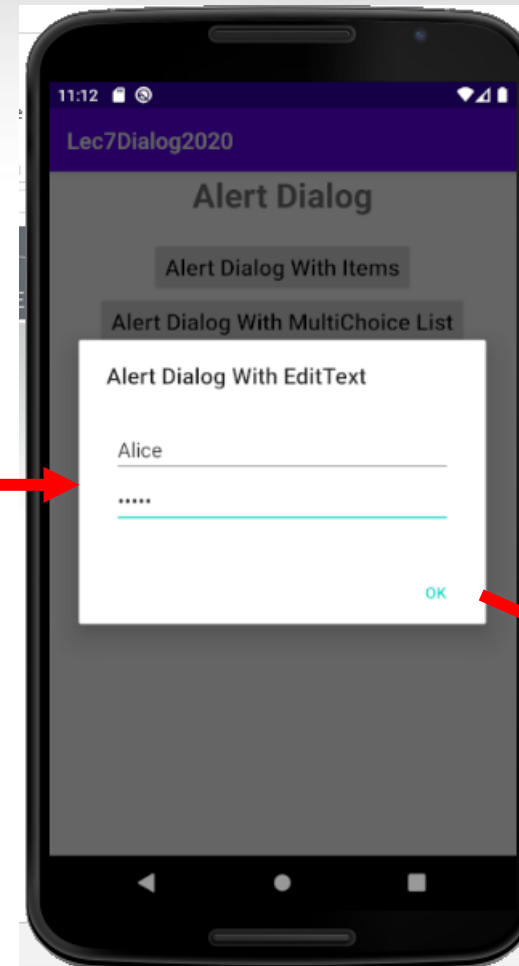
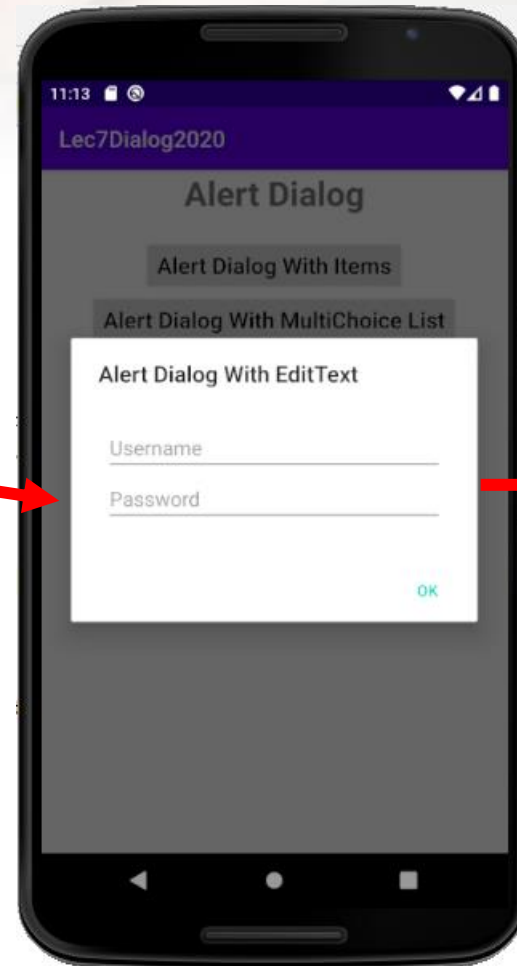
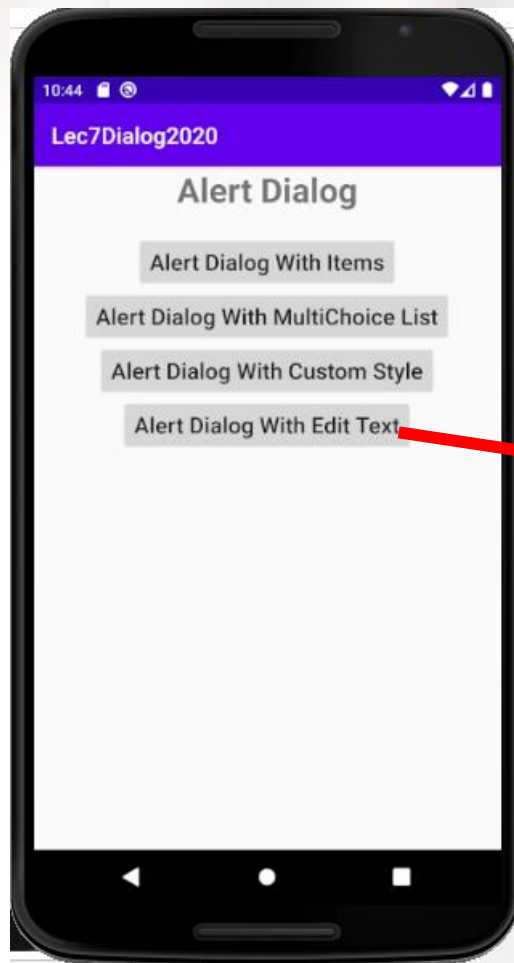
```
<style name="AlertDialogCustom"
parent="@android:style/Theme.Material.Dialog">
    <item name="android:textColor">@android:color/white</item>
    <item name="android:textStyle">bold</item>
    <item name="android:background">@android:color/holo_blue_dark</item>
</style>
```

Alert Dialog With Custom Style : MainActivity.kt

```
fun withCustomStyle(v: View){  
    val myBuilder = AlertDialog.Builder(ContextThemeWrapper(this, R.style.AlertDialogCustom))  
    myBuilder.apply {  
        setTitle("Custom Style Dialog")  
        setMessage("The message from a custom style dialog")  
        setPositiveButton("Yes") { dialog, which ->  
            Toast.makeText(applicationContext, "Click on Yes", Toast.LENGTH_SHORT).show()  
        }  
        setNegativeButton("No") { dialog, which ->  
            Toast.makeText(applicationContext, "Click on No", Toast.LENGTH_SHORT).show()  
        }  
        setNeutralButton("Maybe") { _, _ ->  
            Toast.makeText(applicationContext, "Click on Maybe", Toast.LENGTH_SHORT).show()  
        }  
        show()  
    }  
}
```



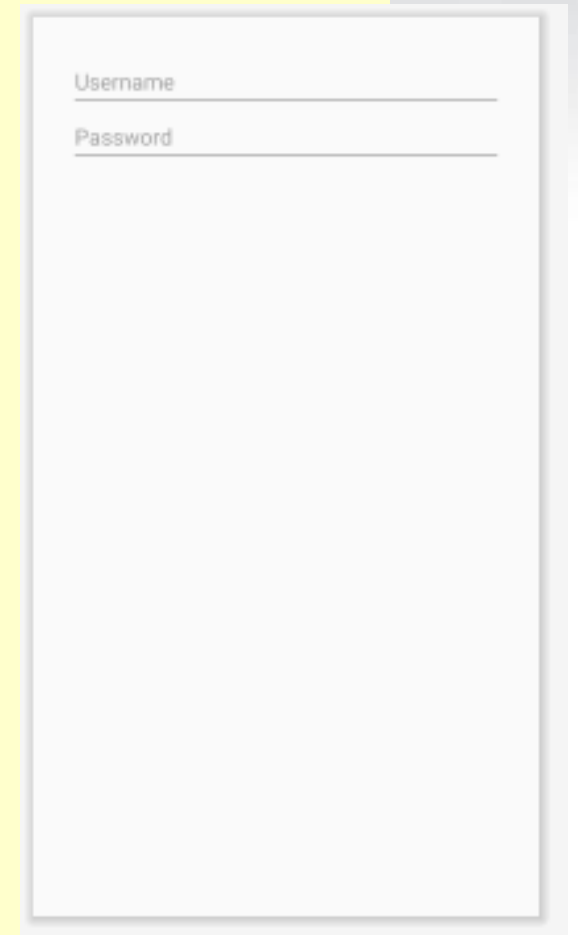
Alert Dialog With Edit Text



Alert Dialog With Edit Text

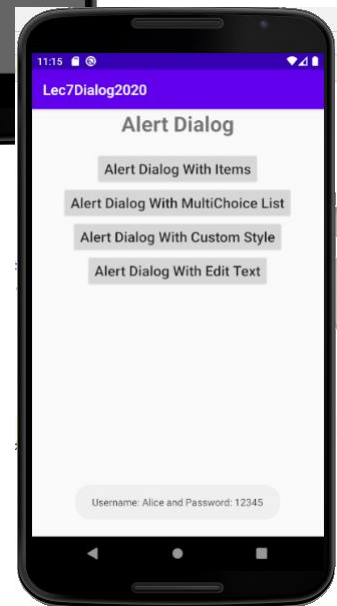
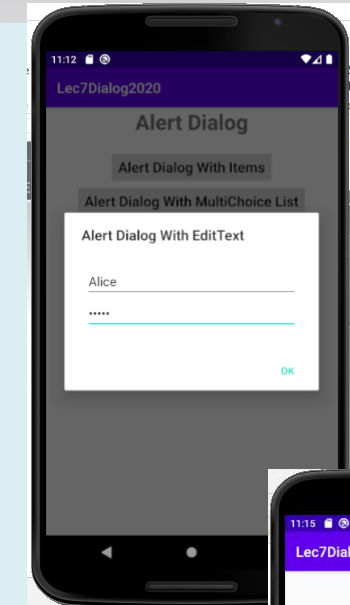
alrtdialog_with_edittext_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/editText_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:hint="Username"/>
    <EditText
        android:id="@+id/editText_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="Password"/>
</LinearLayout>
```



Alert Dialog With Edit Text: MainActivity.kt

```
fun withEditText(v: View){  
    val inflater = LayoutInflater  
    val dialogLayout = inflater.inflate(R.layout.alertdialog_with_edittext_layout, null)  
    val editTextName = dialogLayout.findViewById<EditText>(R.id.editText_name)  
    val editTextPassword = dialogLayout.editText_password  
    val myBuilder = AlertDialog.Builder(this)  
    myBuilder.apply {  
        setTitle("Alert Dialog With EditText")  
        setView(dialogLayout)  
        setPositiveButton("OK") { _, _ -> Toast.makeText(applicationContext, "Username: "  
            + editTextName.text.toString() + " and Password: " + editTextPassword.text.toString(),  
            Toast.LENGTH_SHORT).show()  
        }  
        show()  
    }  
}
```





RecyclerView

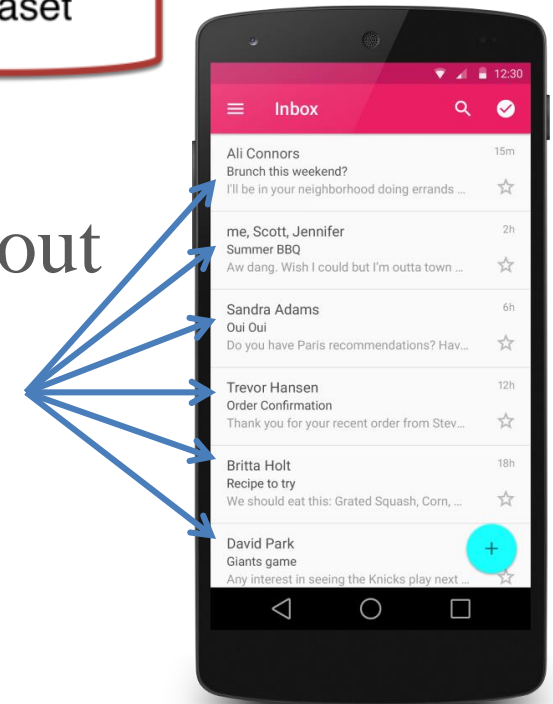
- More advanced and flexible version of ListView.
- This widget is a container for displaying large data sets that can be scrolled very efficiently by maintaining a limited number of views.
 - Provides built
 - Layout manages for positions
 - Default animations for common item operations, such as removal or addition of items

RecyclerView

- Like the listview, uses an adapter



- Normally use a cardview as the default layout for each item in the list.
 - Similar to a fragment.





RecyclerView

RecyclerView requires

- a **set of data objects** to work with
- an **adapter** to bind that data to the views shown in the ViewHolders
- an **xml file of the individual view item**
- a **ViewHolder** to populate the UI from the xml item file
- **LayoutManager** : LinearLayoutManager, GridLayoutManager, Staggered Grid Layout Manager
- ItemDecoration Or CardView

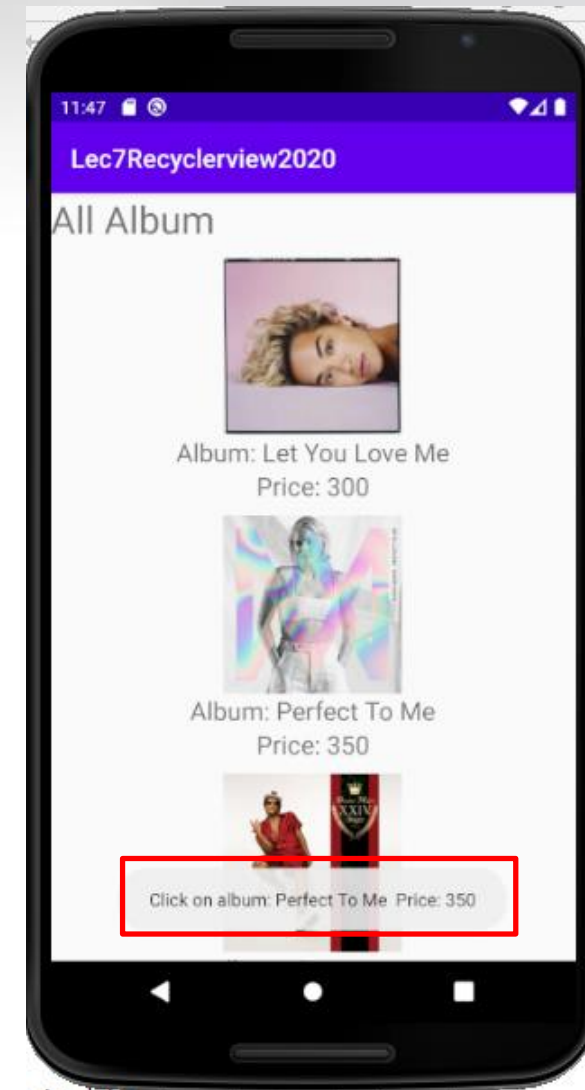
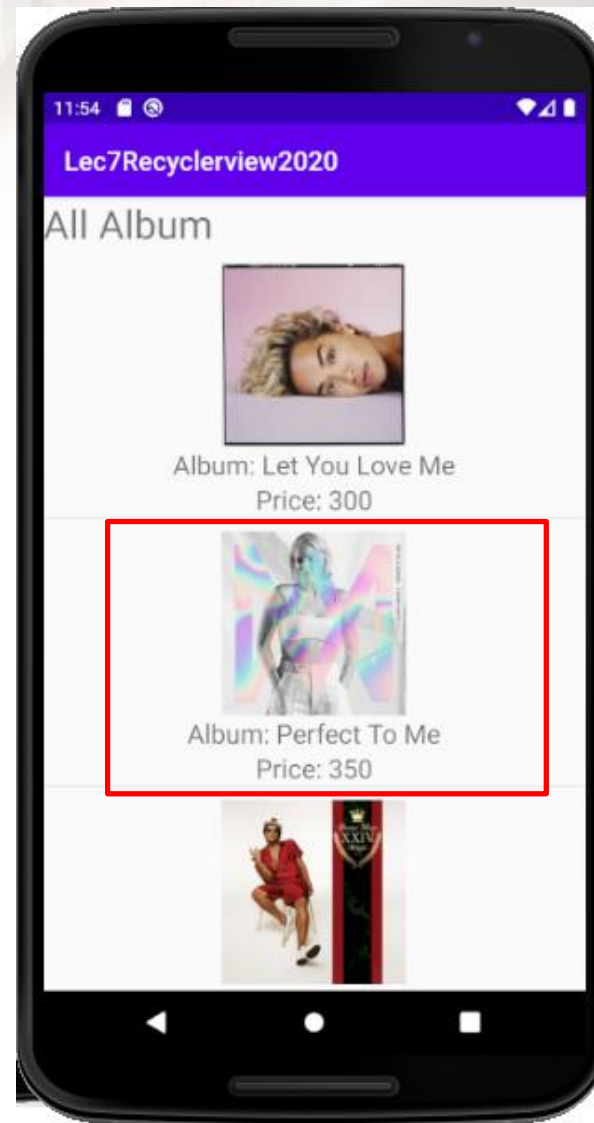


RecyclerView

Using a RecyclerView has the following key steps:

- Add RecyclerView **support library** to the **gradle build file**
- Add a RecyclerView to **activity to display** the items
- Define **a model class** to use as the data source
- Create a **custom row layout XML file** to visualize the item
- Create a **RecyclerView.Adapter** and **ViewHolder** to render the item
- **Bind the adapter to the data source** to populate the RecyclerView
- Add RecyclerView **Item Click Listener**

RecyclerView : Example





Set RecyclerView

- Use RecyclerView Library at build.gradle

```
implementation 'androidx.recyclerview:recyclerview:1.1.0'
```

- Use glide library to show image

```
implementation 'com.github.bumptech.glide:glide:4.9.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'
```

- For use internet : App>> manifests >> Androidmanifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```



RecyclerView

- `build.gradle(module: app)`

Add : `implementation 'com.github.bumptech.glide:glide:4.11.0'`
`annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'`

```
dependencies {  
    implementation fileTree(dir: "libs", include: ["*.jar"])  
    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"  
    implementation 'androidx.core:core-ktx:1.3.1'  
    implementation 'androidx.appcompat:appcompat:1.2.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.1'  
    implementation 'androidx.recyclerview:recyclerview:1.1.0'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    implementation 'com.github.bumptech.glide:glide:4.11.0'  
    annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'  
}
```

RecyclerView

- For use internet : App>> manifests >> Androidmanifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```





Recycler View: File

- activity_main.xml
- album_item_layout.xml
- Class: ItemAlbum.kt
- Class: MyViewHolder.kt
- Class: AlbumAdapter.kt
- MainActivity.kt

RecyclerView

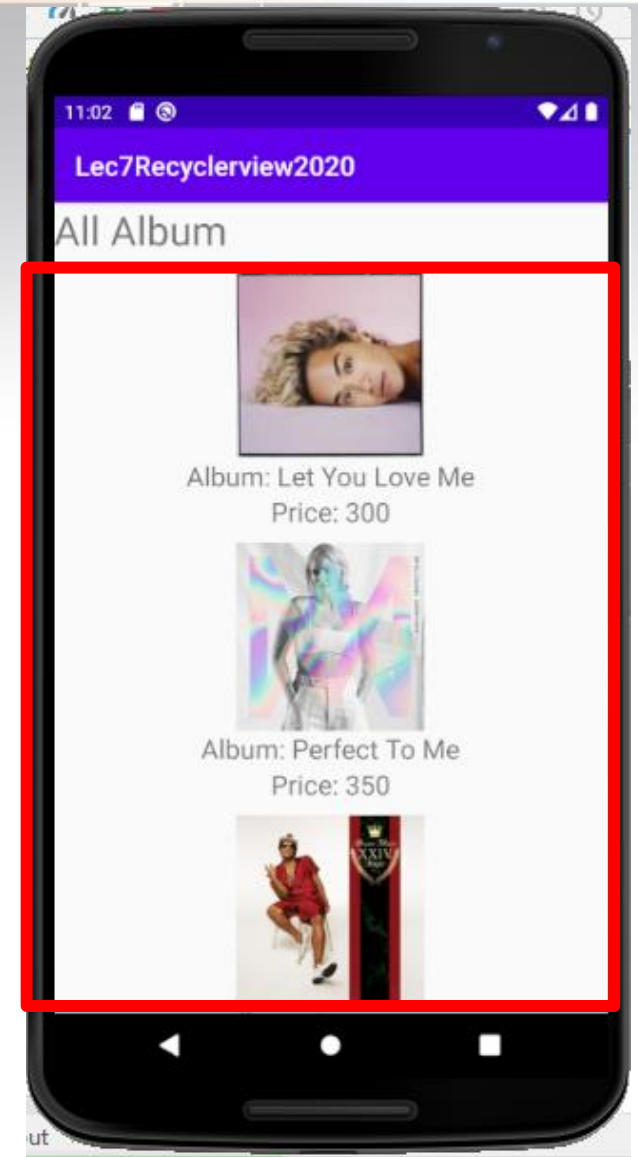
activity_main.xml

```
<?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"
    android:padding="5dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="All Album"
        android:textSize="30sp"/>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical"/>

</LinearLayout>
```

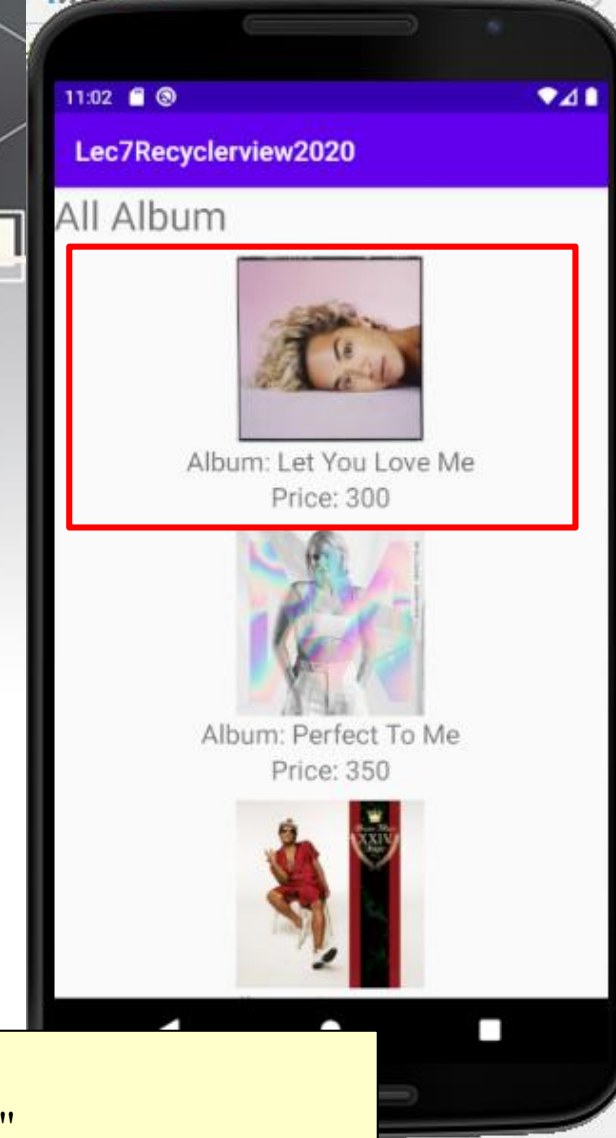


Show data on RecyclerView

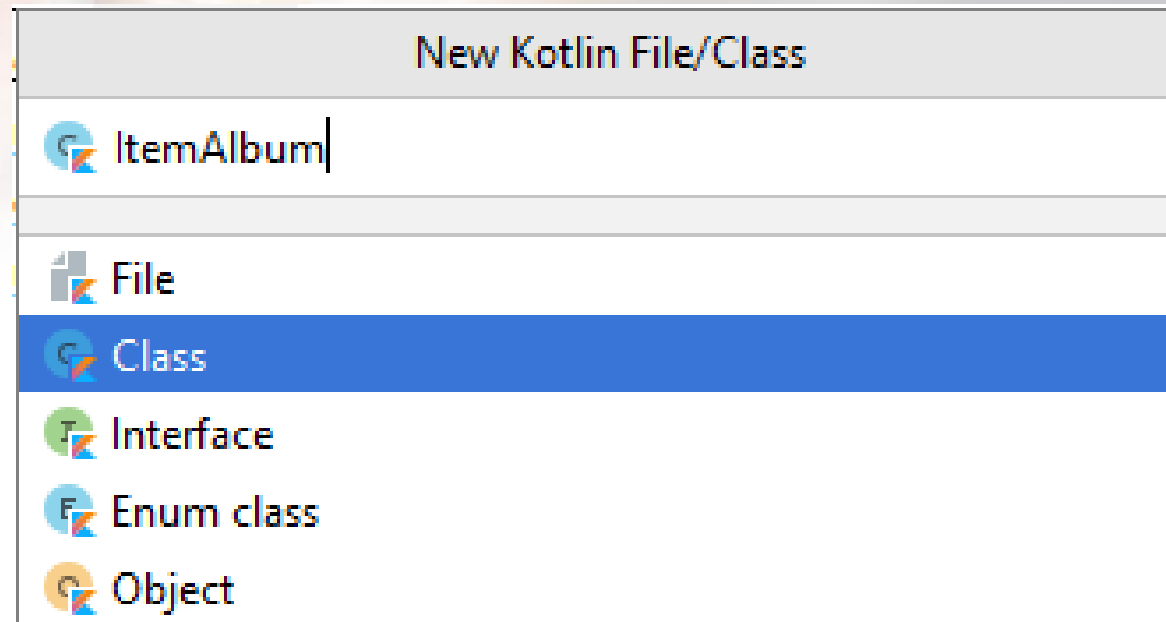
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <ImageView
        android:id="@+id/imageAlbum"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:paddingTop="10dp"
        android:layout_centerHorizontal="true"
        android:src="@drawable/harry"/>
    <TextView
        android:id="@+id/nameText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Test"
        android:textSize="20sp"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/imageAlbum"/>
```

album_item_layout.xml

```
<TextView
    android:id="@+id/priceText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Test"
    android:textSize="20sp"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/nameText"/>
</RelativeLayout>
```



Create Class : ItemAlbum.kt

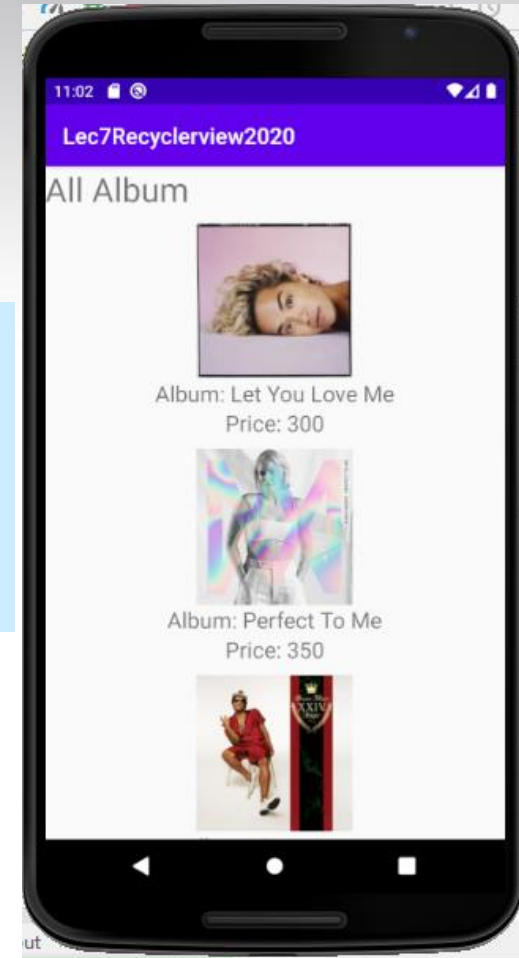


- ItemAlbum.kt

```
data class ItemAlbum(var name:String?, var price: Int?, var image: String?) {  
}
```

Create Class: MyViewHolder.kt

```
class MyViewHolder (view: View) : RecyclerView.ViewHolder(view) {  
    var tvName = view.nameText  
    var tvPrice = view.priceText  
    var mvImage = view.imageAlbum }  
}
```



Create Adapter : AlbumAdapter.kt

```
class AlbumAdapter(val items : List<ItemAlbum>, val context: Context) : RecyclerView.Adapter<MyViewHolder>(){

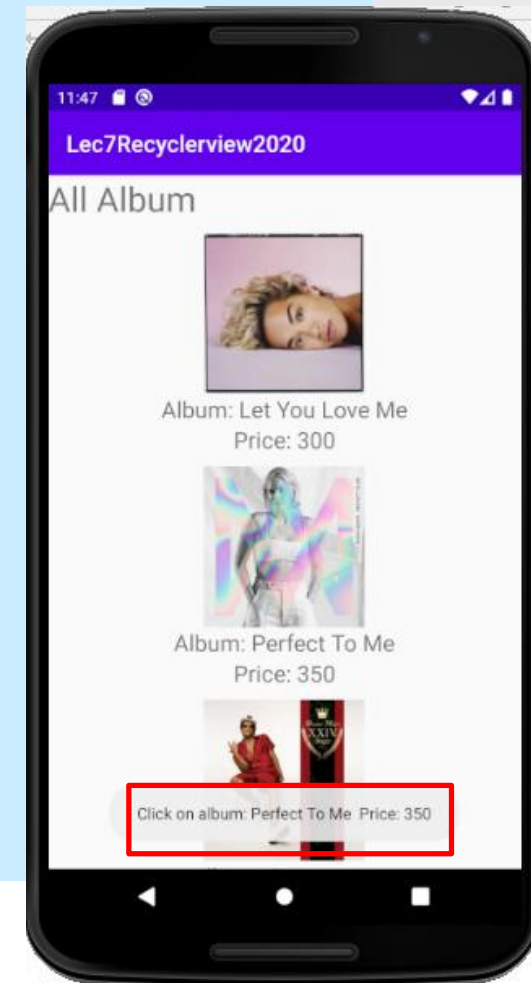
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {
        val view_item = LayoutInflater.from(parent.context).inflate(R.layout.album_item_layout, parent, false)

        val myHolder = MyViewHolder(view_item)
        myHolder.itemView.setOnClickListener(){

            val pos = myHolder.adapterPosition
            Toast.makeText(context, "Click on album: ${items[pos].name} Price: ${items[pos].price}",
                Toast.LENGTH_SHORT).show()

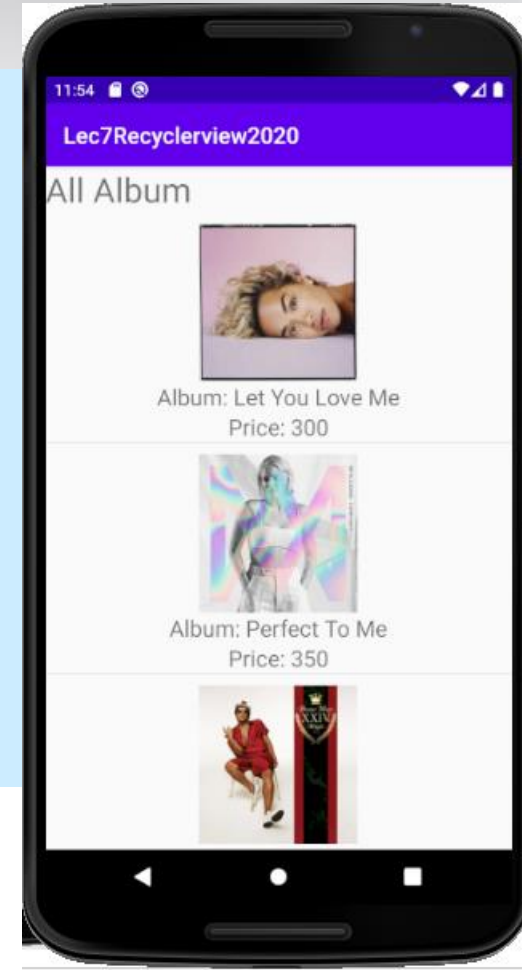
        }

        return myHolder
    }
}
```



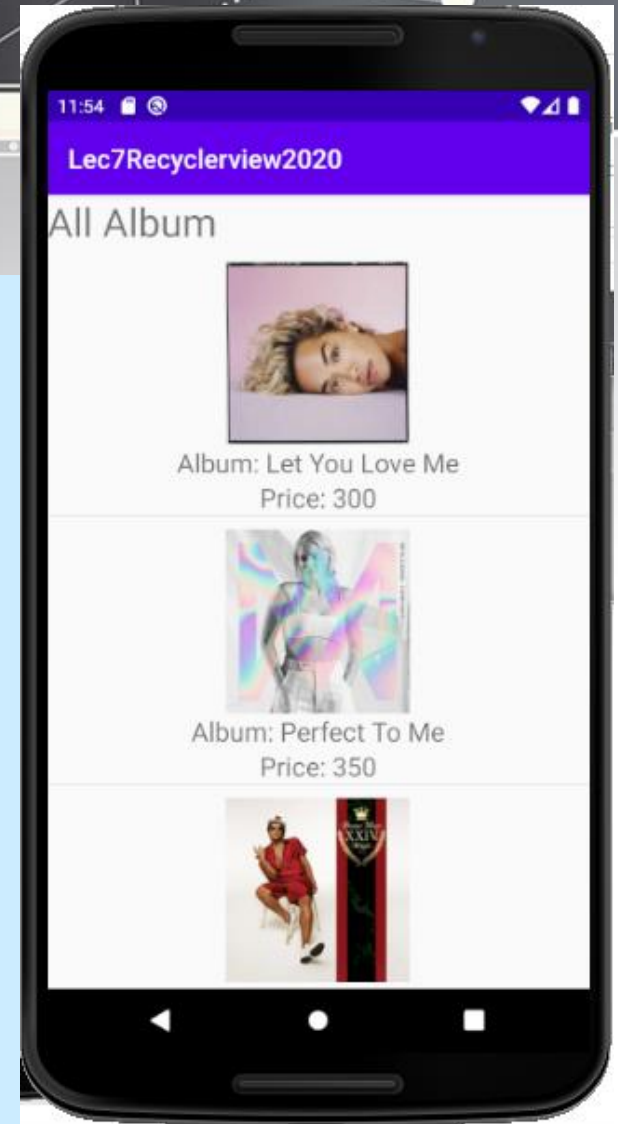
Create Adapter :AlbumAdapter.kt (cont.)

```
override fun onBindViewHolder(holder: MyViewHolder, position: Int) {  
  
    holder.tvName?.text = "Album: " + items[position].name  
    holder.tvPrice?.text = "Price: " + items[position].price.toString()  
    Glide.with(context).load(items[position].image).into(holder.mvImage)  
  
}  
  
override fun getItemCount(): Int {  
    return items.size  
}  
}
```



MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
    val albumList= arrayListOf<ItemAlbum>()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        prepareAlbum()  
        recycler_view.adapter = AlbumAdapter(this.albumList,applicationContext)  
        recycler_view.layoutManager = LinearLayoutManager(applicationContext)  
  
        val itemDecor = DividerItemDecoration(this, DividerItemDecoration.VERTICAL)  
        recycler_view.addItemDecoration(itemDecor)  
    }  
    fun prepareAlbum(){  
        albumList.add(ItemAlbum("Let You Love Me", 300,"https://sv1.picz.in.th/images/2019/09/05/ZrL8Jq.jpg"))  
        albumList.add(ItemAlbum("Perfect To Me",350,"https://sv1.picz.in.th/images/2019/09/05/ZrJ6vk.png"))  
        albumList.add(ItemAlbum("24K Magic",300,"https://sv1.picz.in.th/images/2019/09/05/ZrJC9e.png"))  
    }  
}
```

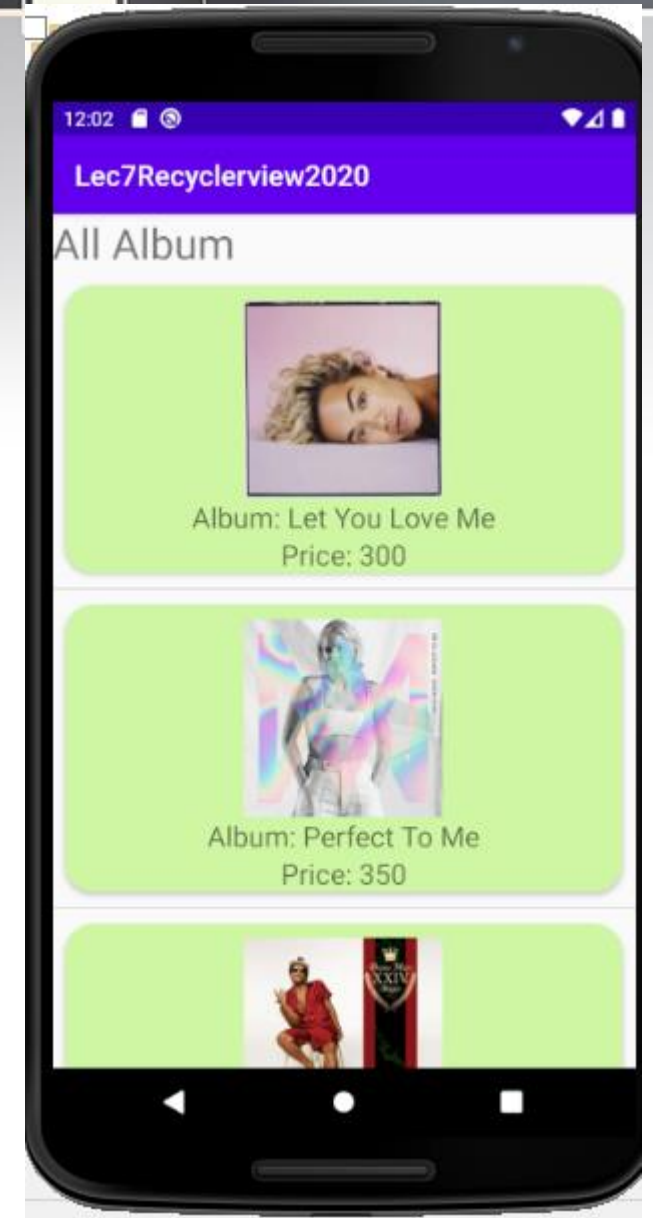


Item Layout: Cardview

- build.gradle(module: app)

implementation 'androidx.cardview:cardview:1.0.0'

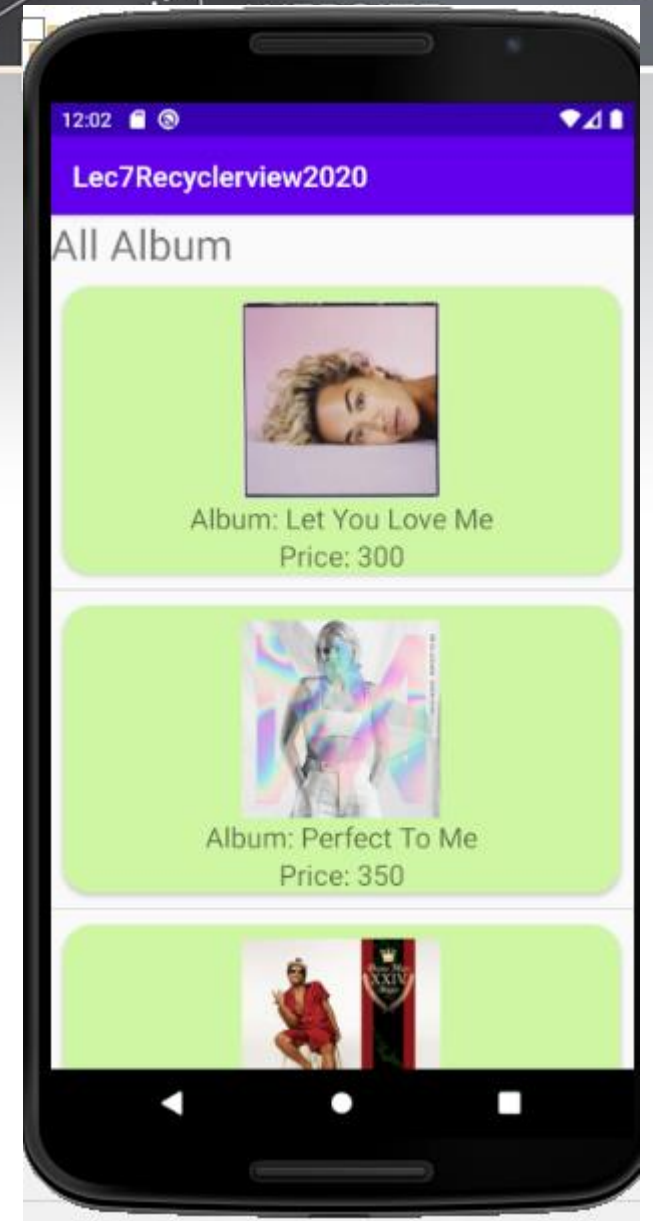
```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    .....
    app:cardBackgroundColor="#CFF6A2"
    app:cardCornerRadius="20dp"
    app:cardElevation="3dp"
    app:cardUseCompatPadding="true">
    <Layout>
        .....
    </Layout>
</androidx.cardview.widget.CardView>
```



Item Layout: Cardview

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
```

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    app:cardBackgroundColor="#CFF6A2"
    app:cardCornerRadius="20dp"
    app:cardElevation="3dp"
    app:cardUseCompatPadding="true">
```



Item Layout: Cardview

<RelativeLayout

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

<ImageView

```
    android:id="@+id/imageAlbum"  
    android:layout_width="150dp"  
    android:layout_height="150dp"  
    android:layout_centerHorizontal="true"  
    android:paddingTop="10dp"  
    android:src="@drawable/harry" />
```

<TextView

```
    android:id="@+id/nameText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/imageAlbum"  
    android:layout_centerHorizontal="true"  
    android:text="Test"  
    android:textSize="20sp" />
```

<TextView

```
    android:id="@+id/priceText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/nameText"  
    android:layout_centerHorizontal="true"  
    android:text="Test"  
    android:textSize="20sp" />
```

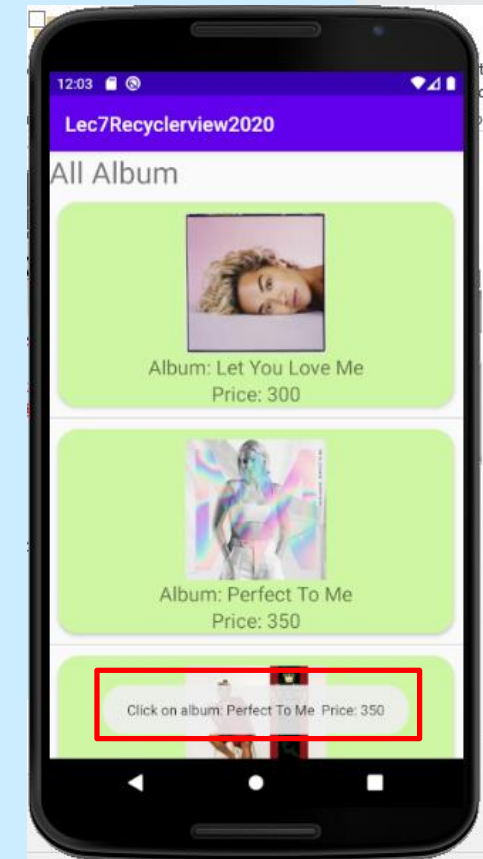
</RelativeLayout>

</androidx.cardview.widget.CardView>

</LinearLayout>

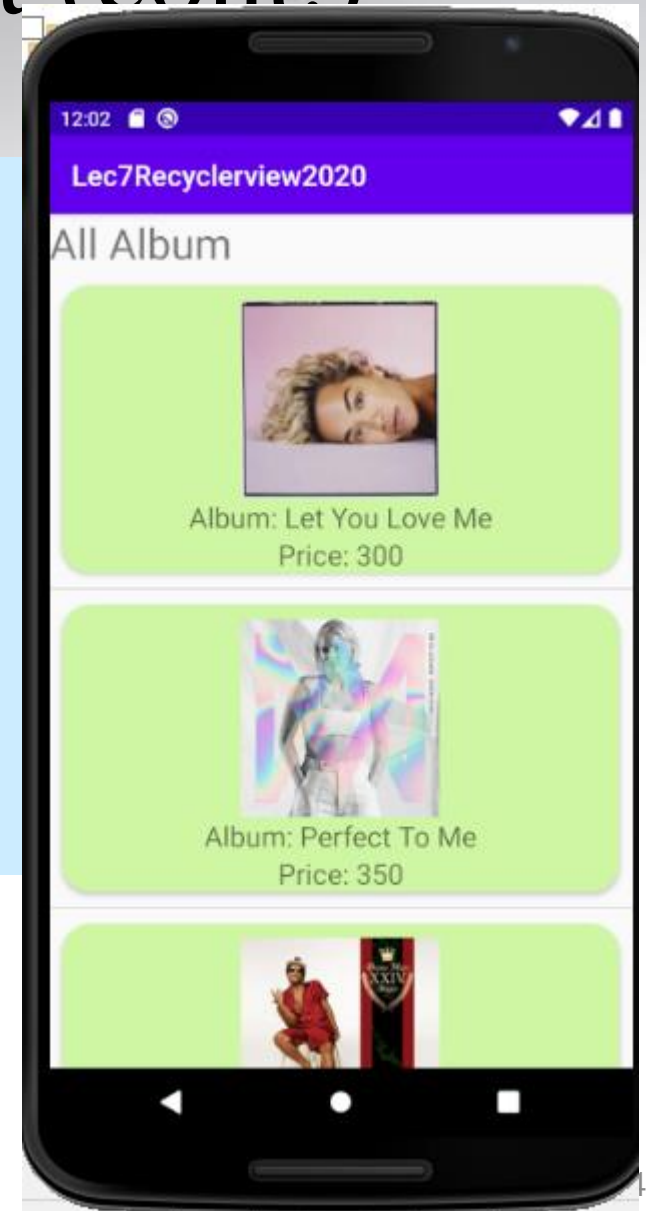
Create Adapter : AlbumAdapter.kt

```
class AlbumAdapter(val items : List<ItemAlbum>, val context: Context) : RecyclerView.Adapter<MyViewHolder>(){  
  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {  
        val view_item = LayoutInflater.from(parent.context).inflate(album_item_cradview_layout, parent, false)  
  
        val myHolder = MyViewHolder(view_item)  
        myHolder.itemView.setOnClickListener(){  
  
            val pos = myHolder.adapterPosition  
            Toast.makeText(context, "Click on album: ${items[pos].name} Price: ${items[pos].price}",  
                Toast.LENGTH_SHORT).show()  
        }  
  
        return myHolder  
    }  
}
```



Create Adapter : AlbumAdapter.kt (cont.)

```
override fun onBindViewHolder(holder: MyViewHolder, position: Int) {  
  
    holder.tvName?.text = "Album: " + items[position].name  
    holder.tvPrice?.text = "Price: " + items[position].price.toString()  
    Glide.with(context).load(items[position].image).into(holder.mv_image))  
  
}  
  
override fun getItemCount(): Int {  
    return items.size  
}  
}
```





References

- https://google-developer-training.gitbooks.io/android-developer-fundamentals-course-concepts/content/en/Unit%204/101_c_sqlite_database.html
- <https://stackoverflow.com/questions/17529766/view-contents-of-database-file-in-android-studio>
- <https://www.androidhive.info/2011/11/android-sqlite-database-tutorial/>
- <https://www.androidhive.info/2016/01/android-working-with-recycler-view/>
- [Dialog https://www.androidly.net/309/android-alert-dialog-using-kotlin](https://www.androidly.net/309/android-alert-dialog-using-kotlin)
- <https://www.simplifiedcoding.net/android-custom-dialog-example/>
- <https://developer.android.com/guide/topics/ui/layout/recyclerview>