# ANDROID PROGRAMMING
# LESSON 4

# Agenda

- Manually Creating an XML Layout
- Managing Constraints using Constraint Sets
- An Overview and Example of Android Event Handling (01)
- Event Handler (02)
- TimePicker, DatePicker (03)
- ListView (04)
- ListView, Spinner (05)
- AutoCompleteTextView, MultiAutocompleteTextView (06)
- Menu (07)
- RecyclerView and CardView (08)(09)

# Manually Creating an XML Layout

```
<?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">
```

# Manually Creating an XML Layout

```
<Button
    android:id="@+id/b1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#1eb729"
    android:text="B1 - HEAD"
    app:layout_constraintBottom_toTopOf="@id/b5"
    app:layout_constraintHorizontal_chainStyle="spread"
    app:layout_constraintRight_toLeftOf="@id/b2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/b4" />
```

# Manually Creating an XML Layout

```
<Button
    android:id="@+id/b2"
    android:text="B2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toRightOf="@id/b1"
    app:layout_constraintRight_toLeftOf="@id/b3"
    app:layout_constraintTop_toTopOf="parent" />
……….
```

# Manually Creating an XML Layout

- **in** – Inches
- **mm** – Millimeters
- **pt** – Points (1/72 of an inch)
- **dp** – Density-independent pixels. An abstract unit of measurement based on the physical density of the device display relative to a 160dpi display baseline.
- **sp** – Scale-independent pixels. Similar to dp but scaled based on the user's font preference.
- **px** – Actual screen pixels. Use is not recommended since different displays will have different pixels per inch. Use *dp* in preference to this unit.

# Manually Creating an XML Layout

- **Manual XML vs. Visual Layout Design**

| Design mode | XML mode |
| --- | --- |
| Quick | Slow |
| Easy | Complex |
| | |

Design mode is more advantage but in case the view not is common view, design mode is useless.

# Managing Constraints using Constraint Sets

- **Java Code vs. XML Layout Files**

```
Button button = new Button(this);
button.setId(View.generateViewId());
ConstraintLayout myLayout = findViewById(R.id.myLayout);
button.setText("Click here");
myLayout.addView(button);
```

# Managing Constraints using Constraint Sets

- **Constraint Sets**

  *constraint sets* are used to control how a view appears relative to its parent view and other sibling views

```
ConstraintSet constraintSet = new ConstraintSet();
constraintSet.connect(button.getId(),ConstraintSet.RIGHT,
        ConstraintSet.PARENT_ID,ConstraintSet.RIGHT);
constraintSet.connect(button.getId(),ConstraintSet.LEFT,
        ConstraintSet.PARENT_ID,ConstraintSet.LEFT);
constraintSet.connect(button.getId(),ConstraintSet.TOP,
        ConstraintSet.PARENT_ID,ConstraintSet.TOP);
constraintSet.connect(button.getId(),ConstraintSet.BOTTOM,
        ConstraintSet.PARENT_ID,ConstraintSet.BOTTOM);
constraintSet.applyTo(myLayout);
```

# Managing Constraints using Constraint Sets

- Set Width and Heigh

```
constraintSet.constrainWidth(button.getId(),
ConstraintSet.WRAP_CONTENT);
constraintSet.constrainHeight(button.getId(),
ConstraintSet.WRAP_CONTENT);
```

# Managing Constraints using Constraint Sets

- Set margin

```
constraintSet.connect(button.getId(), ConstraintSet.LEFT,
        ConstraintSet.PARENT_ID, ConstraintSet.LEFT, 200);
constraintSet.connect(button.getId(), ConstraintSet.TOP,
        ConstraintSet.PARENT_ID, ConstraintSet.TOP, 200);
```

- Bias constraint

```
constraintSet.setHorizontalBias(button.getId(), 0.25f);
constraintSet.setVerticalBias(button.getId(), 0.25f);
```

# An Overview and Example of Android Event Handling

- **Understanding Android Events**
  - Generated in response to an external action
  - Android framework maintains an ***event queue*** into which events are placed as they occur
  - In order to handle the event, the view must have in place an ***event listener***

# An Overview and Example of Android Event Handling

- **Using callback declaration**

```
button.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        //--------------------TO DO ------------------------

    }

});
```

# An Overview and Example of Android Event Handling

- **Using the *android:onClick* Resource**

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="buttonClick"
    android:text="Click me" />
```

# An Overview and Example of Android Event Handling

- **Event Listeners and Callback Methods**
  - `onClickListener()`
  - `onLongClickListener()`
  - `onTouchListener()`
  - `onFocusChangeListener()`
  - `onKeyListener()`

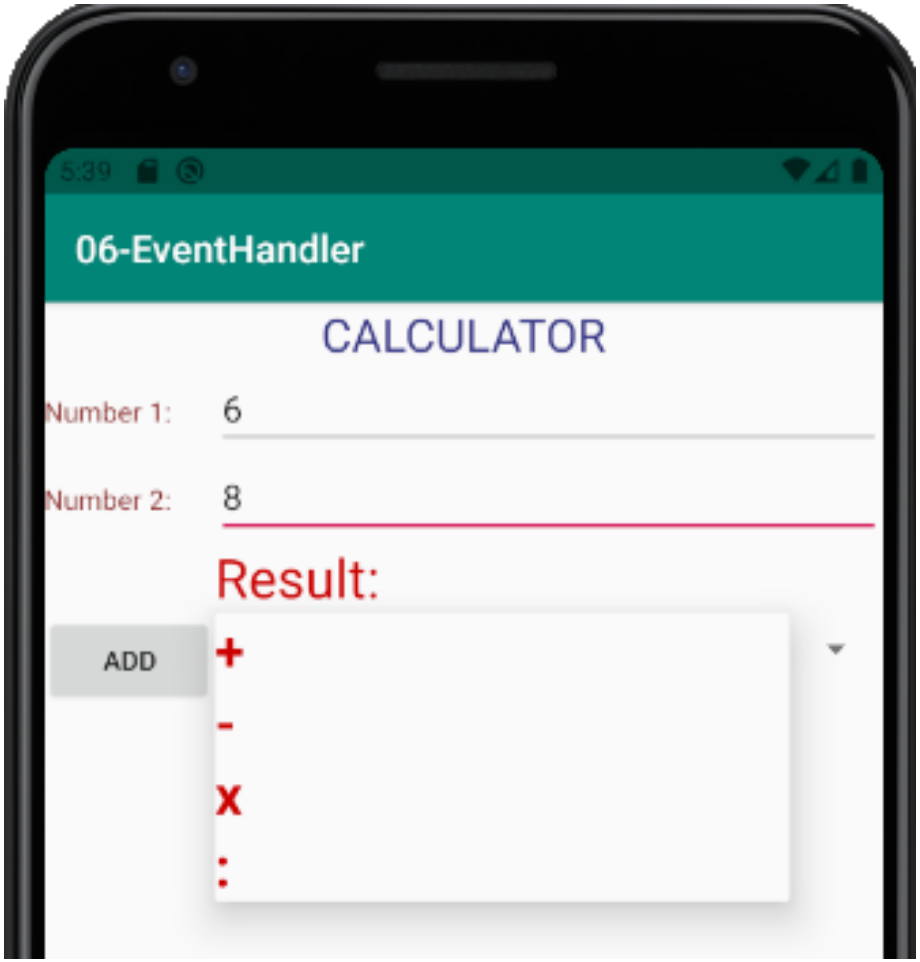
# An Overview and Example of Android Event Handling

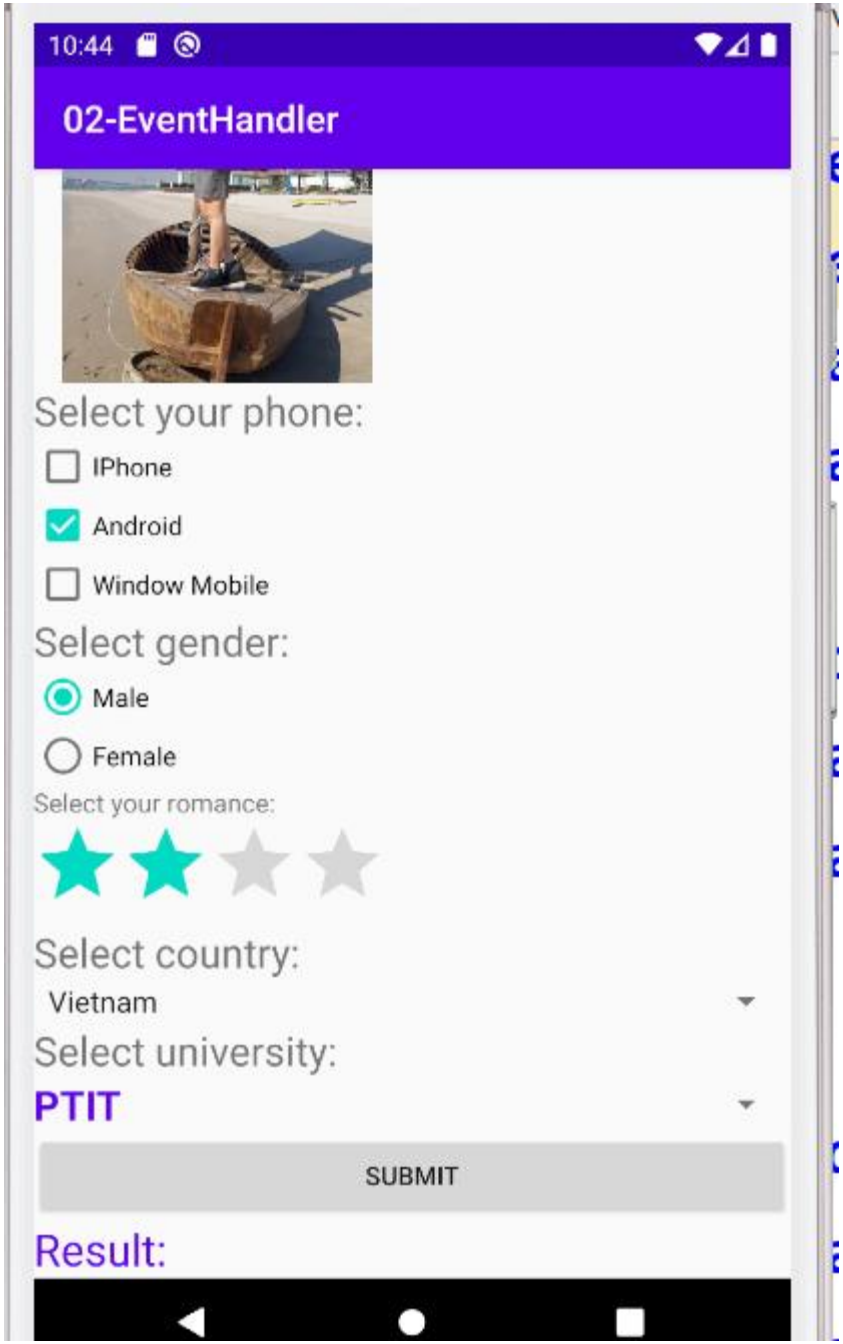
- **Consuming Events**

  How android system manage if one view

```
button.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        TextView editText = findViewById(R.id.editText);
        TextView text = findViewById(R.id.textView);
        text.setText("Hello " + editText.getText());
        Toast.makeText(MainActivity.this, "Hello " + editText.getText(),
Toast.LENGTH_LONG).show();
    }
});
button.setOnLongClickListener(
        new Button.OnLongClickListener() {
            public boolean onLongClick(View v) {
                TextView statusText =
                        (TextView)findViewById(R.id.editText);
                statusText.setText("Long button click");
                return true;
            }
        }
);
```

# An Overview and Example of Android Event Handling (01)





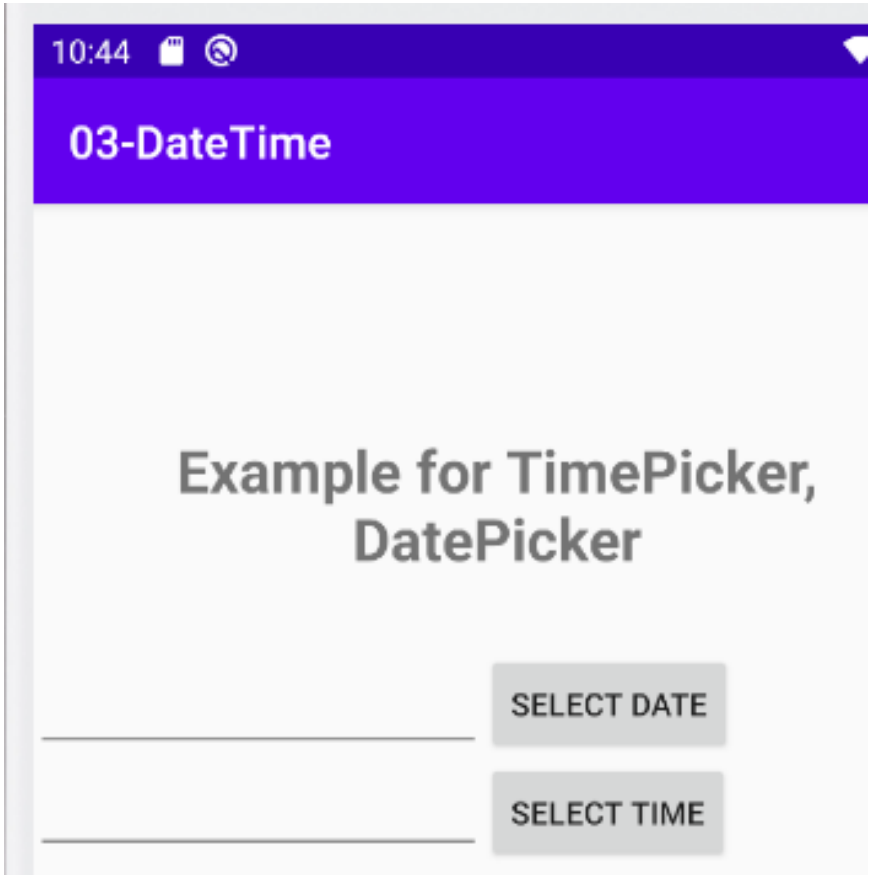

# Event Handler (02)

# An Overview and Example of Android Event Handling

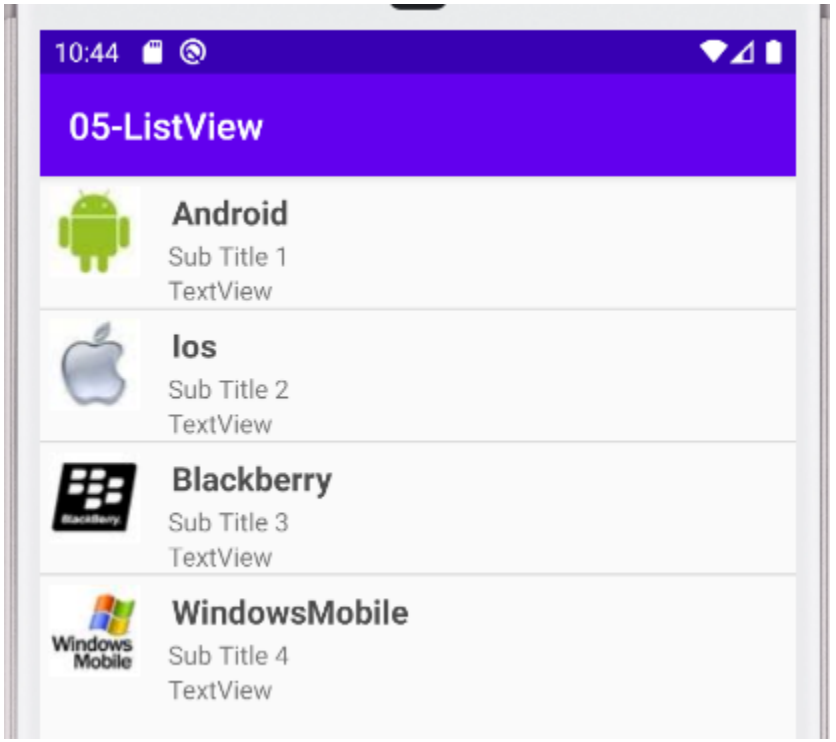
- onFocusChangeListener()

```
final EditText editText = findViewById(R.id.editText);
editText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
        if(hasFocus){
            if(((EditText) v).getText().toString().equals("Name")){
                ((EditText) v).setTextColor(Color.argb(255,0,0,0));
                ((EditText) v).setText("");
            }
        }else {
            if(((EditText) v).getText().toString().isEmpty()){
                ((EditText) v).setText("Name");
                ((EditText) v).setTextColor(Color.argb(255,200,200,200));
            }
        }
    }
});
```

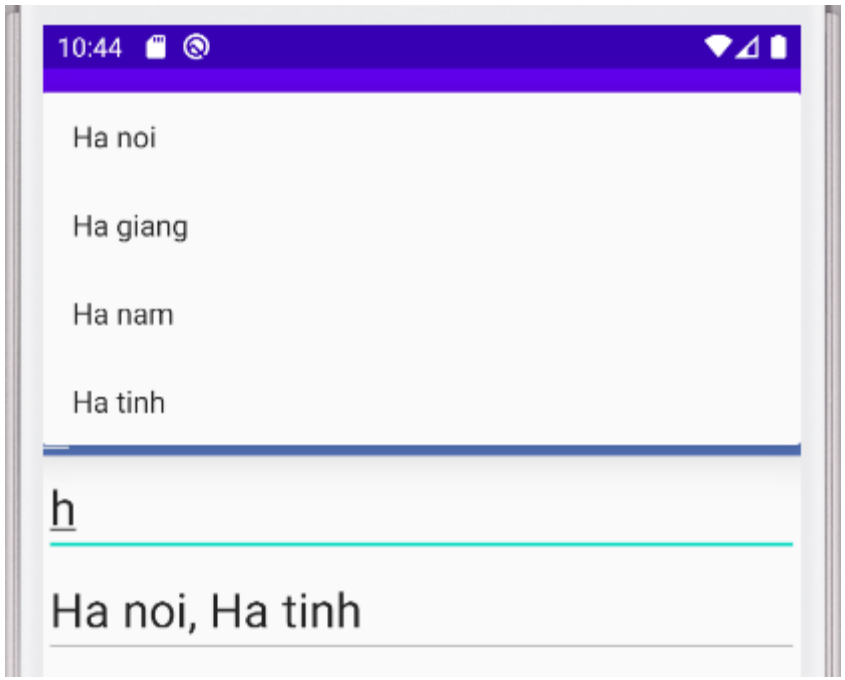
# TimePicker, DatePicker (03)

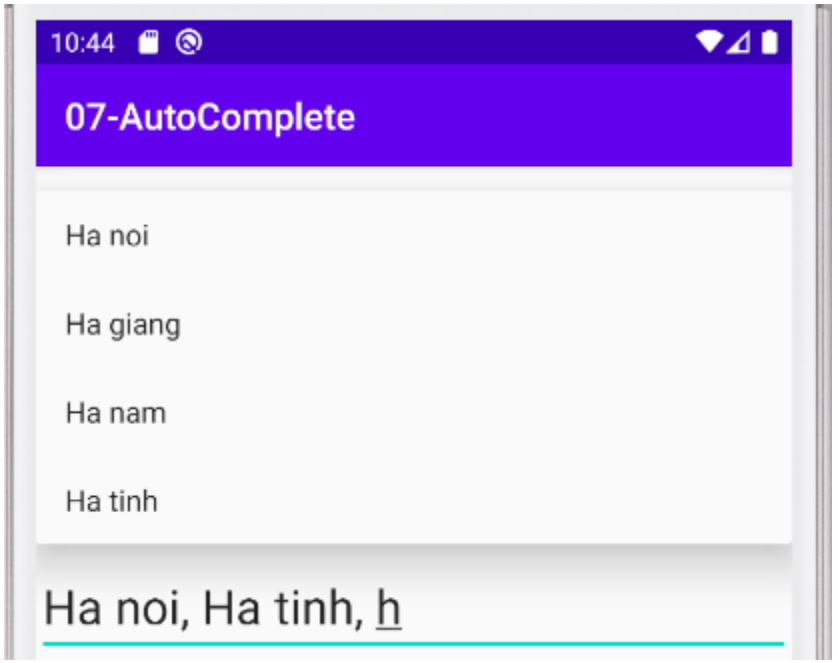


# ListView (04,05)

# AutoCompleteTextView, MultiAutocompleteTextView (06)



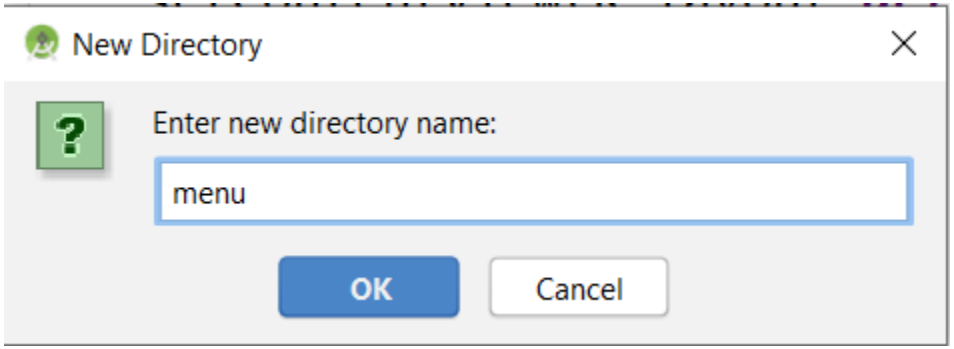



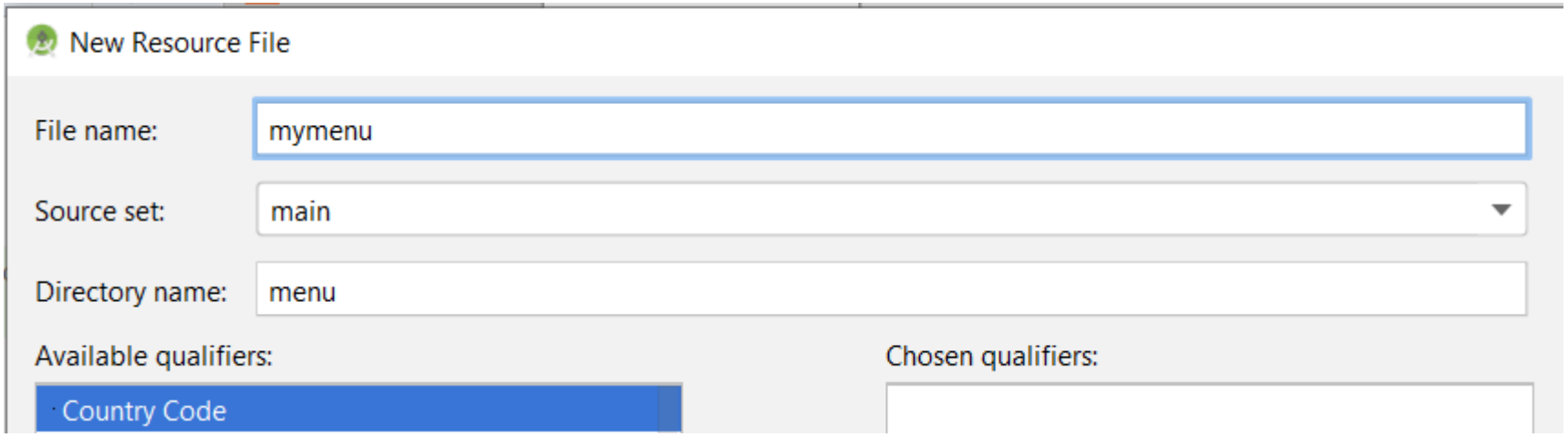
```
<AutoCompleteTextView
    android:id="@+id/oneauto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:completionThreshold="1"
    android:textSize="26dp" >
    <requestFocus />
</AutoCompleteTextView>
<MultiAutoCompleteTextView
    android:id="@+id/multiAuto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:completionThreshold="1"
    android:textSize="26dp" />
```

android:completionThreshold="1“: Mục đích là thiết lập số ký tự bắt đầu lọc trong AutoComplete

# Menu (07)

- Create : a menu directory
  - res/new/Directory
- Create: xml file
  - menu/New/Menu resource file



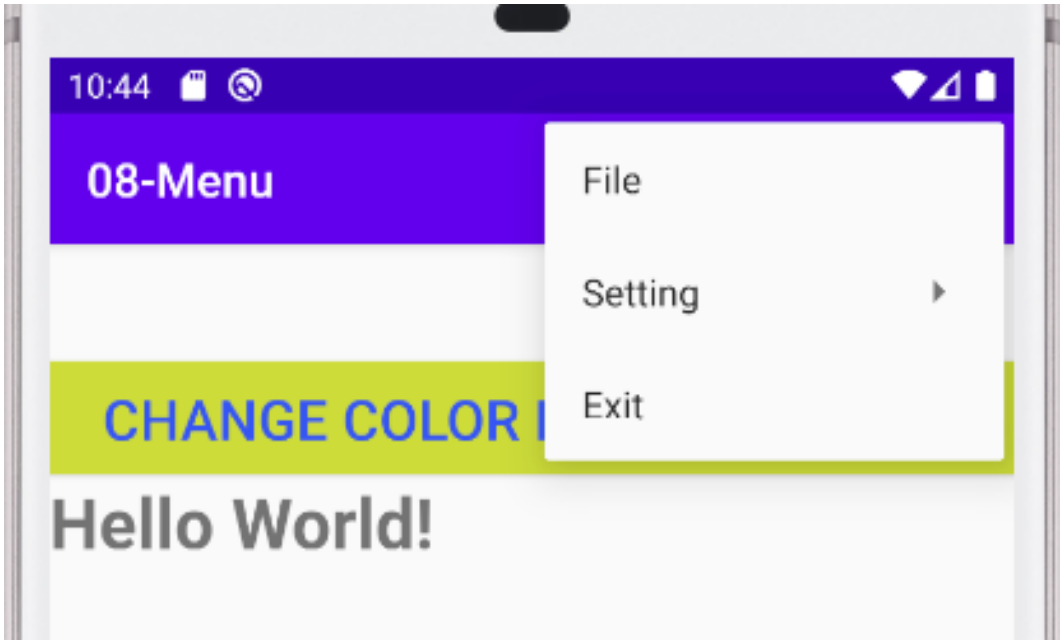



# mymenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/
        apk/res/android">
    <item android:title="File"
        android:id="@+id/mFile"/>
    <item android:title="Contact"
        android:id="@+id/mContact">
        <menu>
            <item android:title="Email"
                android:id="@+id/mEmail"/>
            <item android:title="Phone"
                android:id="@+id/mPhone"/>
        </menu>
    </item>
    <item android:title="Exit"
        android:id="@+id/mExit"/>
</menu>
```

# MainActivity.java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.mymenu,menu);
    return super.onCreateOptionsMenu(menu);
}
```

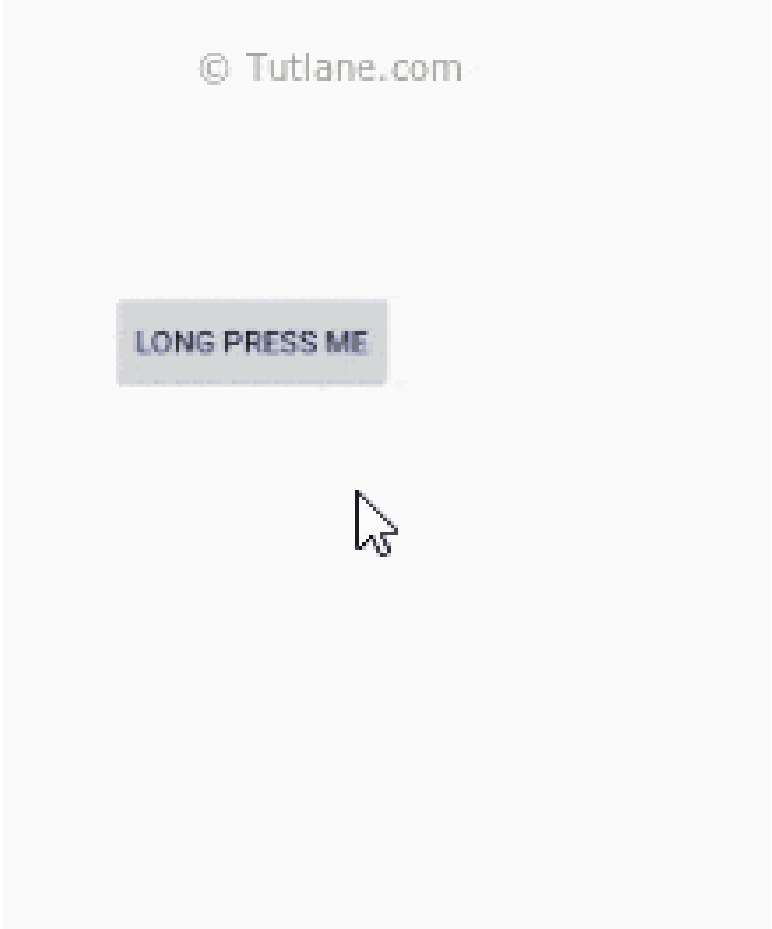


# Select MenuItem

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch(item.getItemId()){
        case R.id.mFile:
            Toast.makeText(this,"Selected File",
                    Toast.LENGTH_LONG).show();
            break;
        case R.id.mExit:
            System.exit(0);
            break;
        case R.id.mEmail:
            Toast.makeText(this,"Seelected Email",
                        Toast.LENGTH_LONG).show();
            break;
        case R.id.mPhone:
            Toast.makeText(this,"Selected phone",
                    Toast.LENGTH_LONG).show();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

# Context Menu

- The android **Context Menu** is more like the menu which displayed on right click in Windows or Linux.
- like a floating menu and that appears when the user performs a long press or click on an element



- Create Context Menu: mycontextmenu.xml

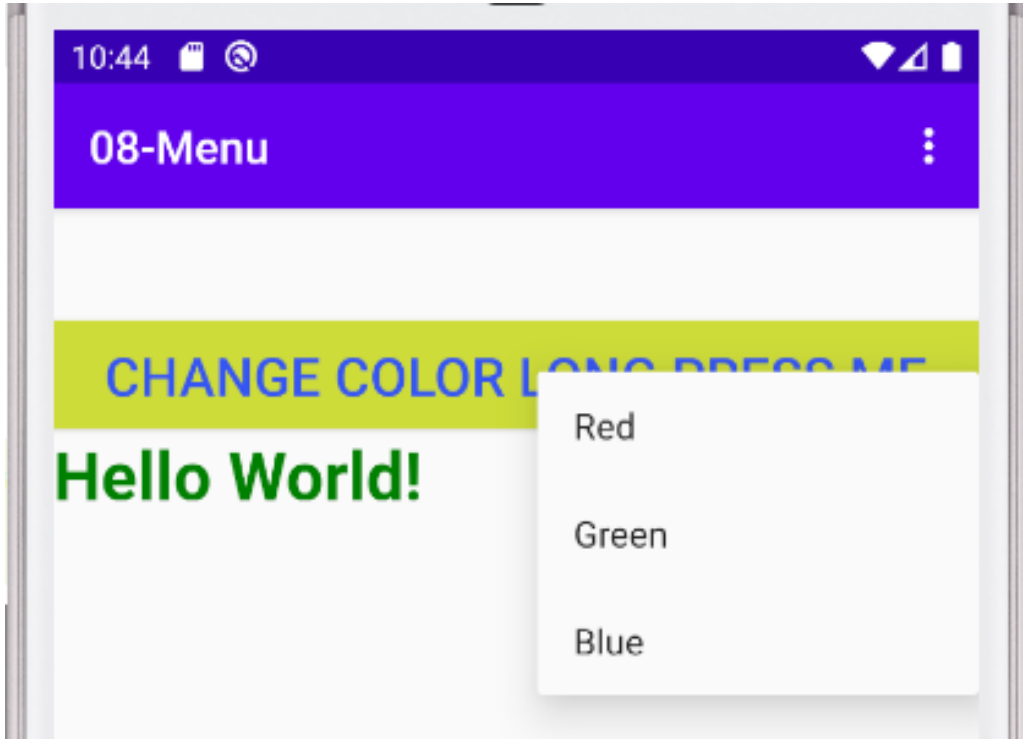
```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/an
droid">
    <item android:title="Red"
        android:id="@+id/mRed"/>
     <item android:title="Green"
         android:id="@+id/mGreen"/>
     <item android:title="Blue"
         android:id="@+id/mBlue"/>
</menu>
```

# Add more colors.xml file

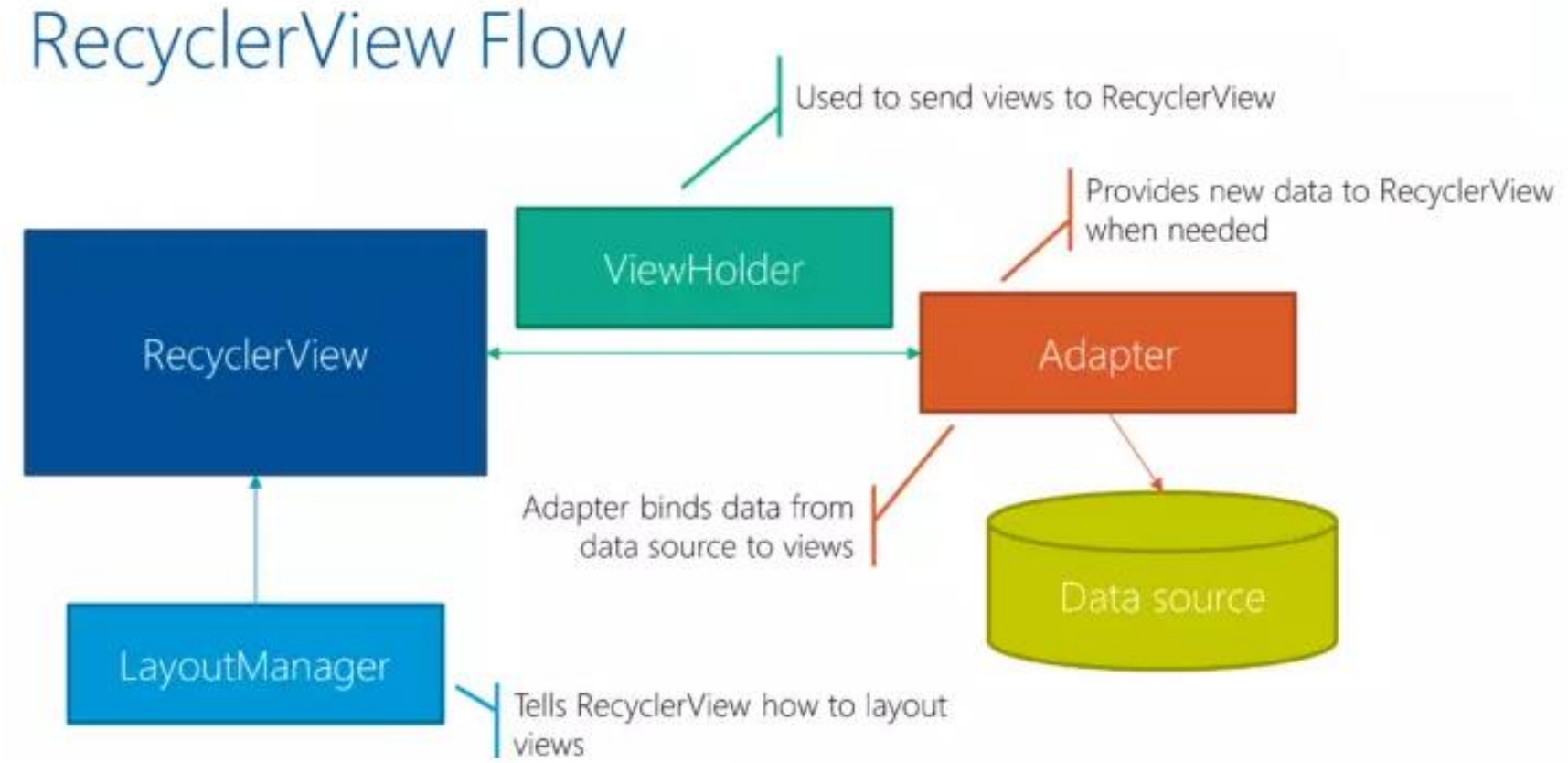
- <color name="cRed">#FF0000</color>
- <color name="cGreen">#008000</color>
- <color name="cBlue">#000080</color>

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    initView();
    registerForContextMenu(bt);
}
private void initView() {
    tv=findViewById(R.id.tv);
    bt=findViewById(R.id.bt);
}
```

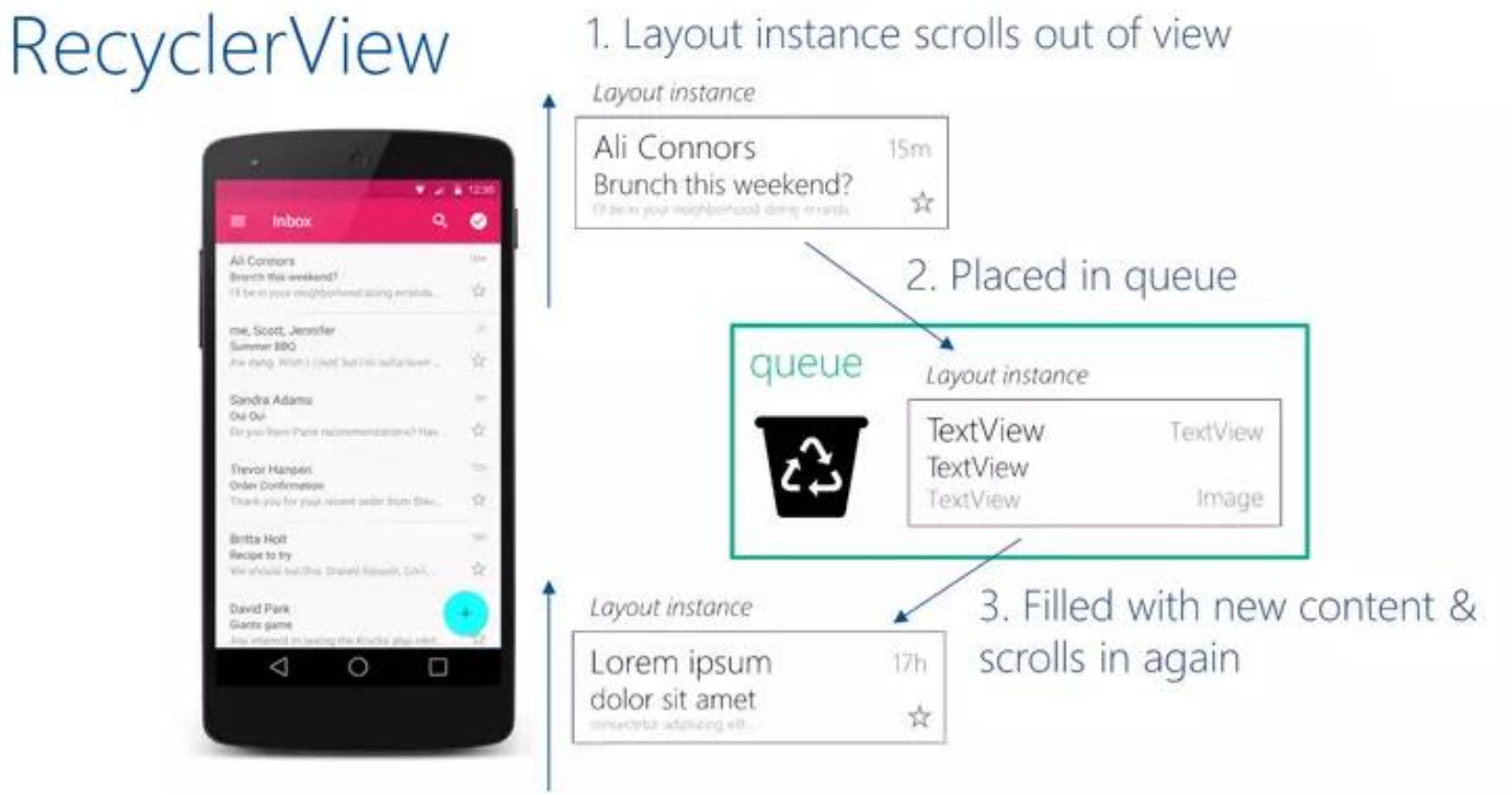
```
@Override
   public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
      super.onCreateContextMenu(menu, v, menuInfo);
      getMenuInflater().inflate(R.menu.mycontextmenu,menu);
   }
   @Override
   public boolean onContextItemSelected(@NonNull MenuItem item) {
      switch(item.getItemId()){
         case R.id.mRed:
            tv.setTextColor(getResources().getColor(R.color.cRed));
            break;
         case R.id.mGreen:
            tv.setTextColor(getResources().getColor(R.color.cGreen));
            break;
         case R.id.mBlue:
            tv.setTextColor(getResources().getColor(R.color.cBlue));
            break;
      }
      return super.onContextItemSelected(item);
   }
```


10:44
08-Menu
CHANGE COLOR L
Hello World!
Red
Green
Blue

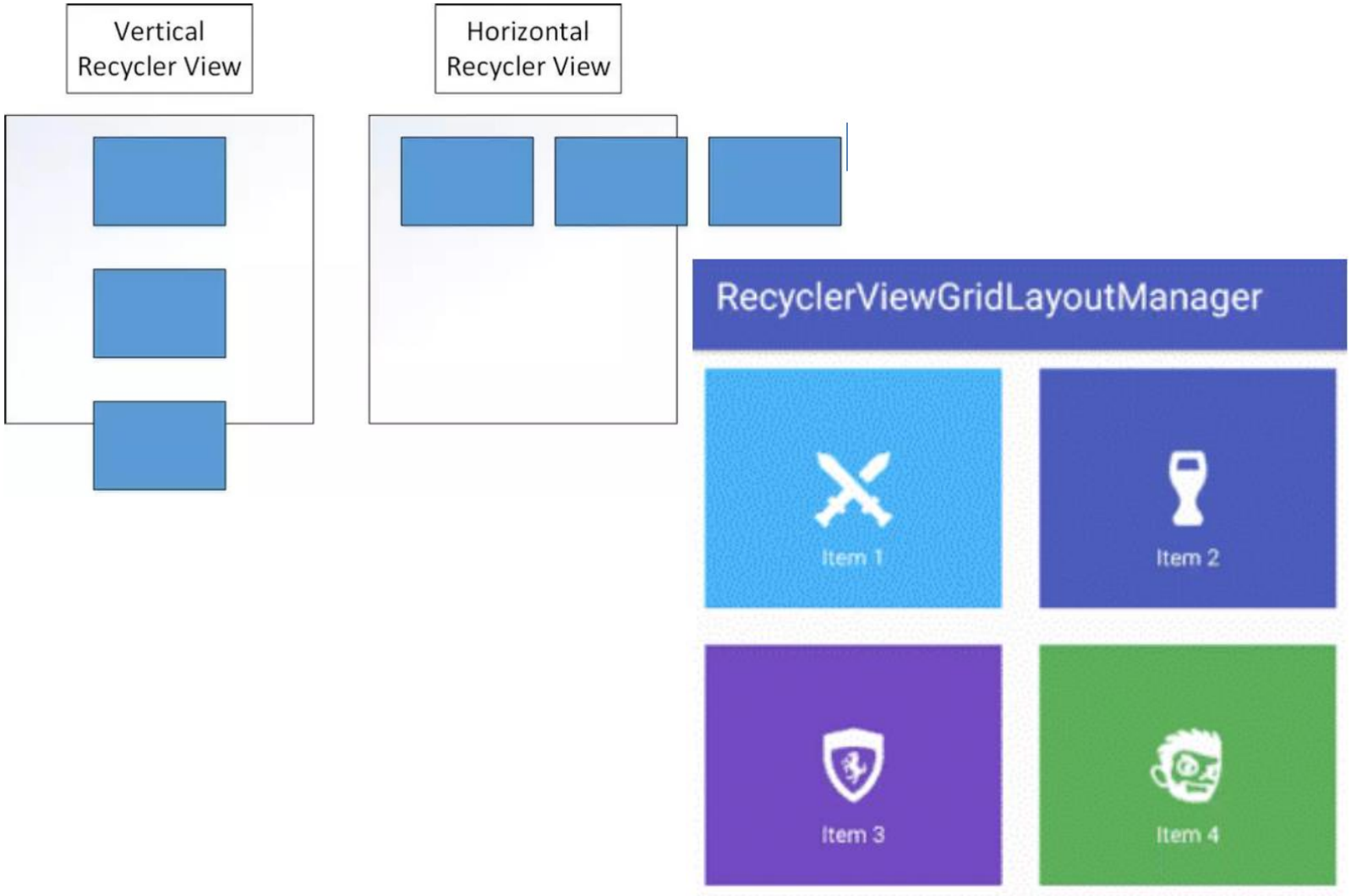
# RecyclerView and CardView (08)



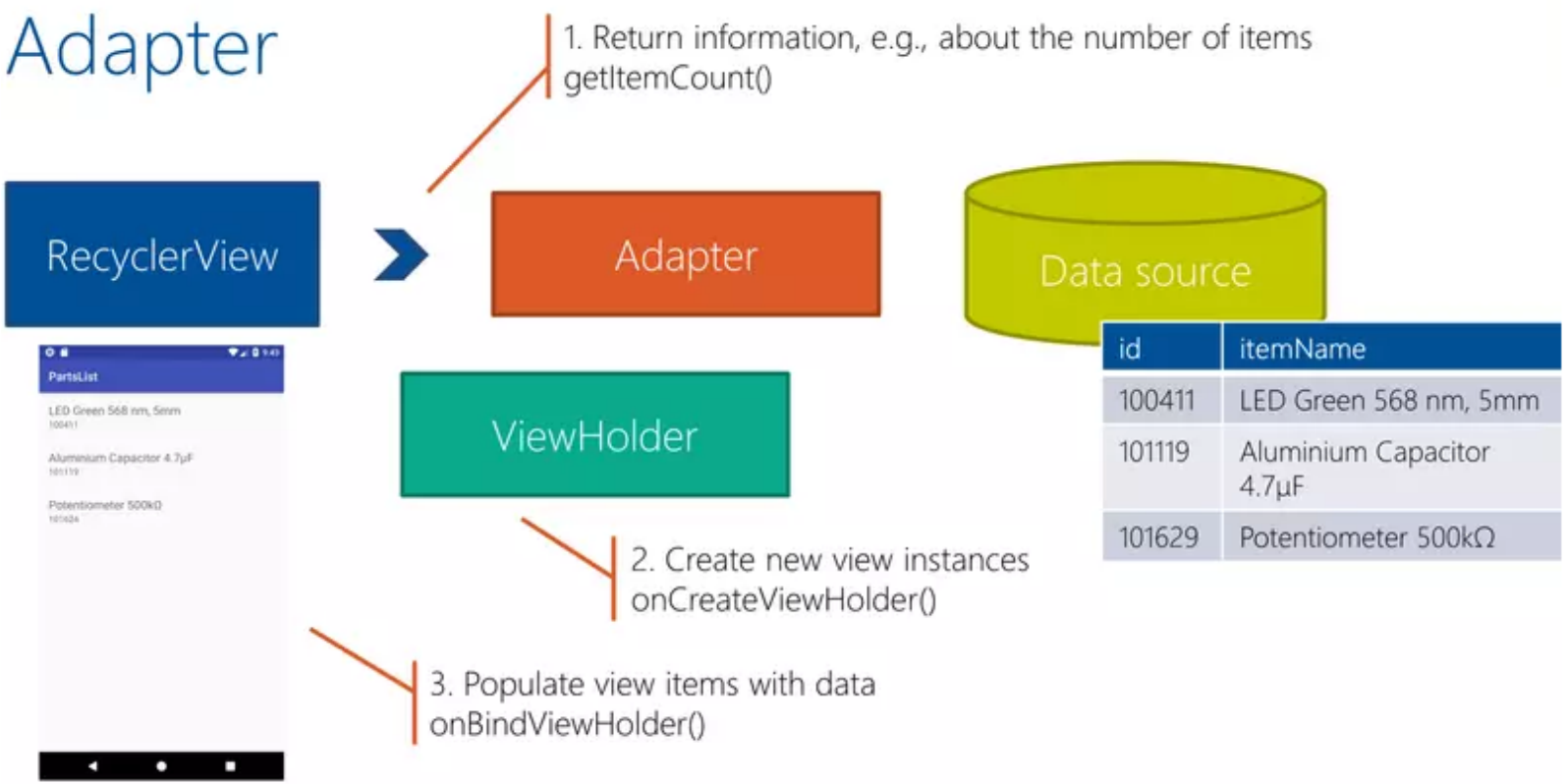
# How does it work?



# LayoutManagers



# RecyclerView.Adapter

# main_activity.xml

```
<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.recyclerview.widget.RecyclerView
        android:id="@+id/rview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="30dp"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

# item.xml

```
<androidx.cardview.widget.CardView
    android:id="@+id/layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardCornerRadius="20dp">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/img"
            android:layout_width="wrap_content"
            android:layout_height="120dp"
            android:src="@drawable/cat1"
            android:scaleType="centerCrop"/>
```

```
<TextView
        android:id="@+id/tv_name"
        android:text="Meo de thuong"
        android:padding="10dp"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:textColor="@color/purple_700"/>
   </LinearLayout>
</androidx.cardview.widget.CardView>
```

# Model class

- Cat.java
- CatViewHolder (extends RecyclerView.ViewHolder)
- CatAdapter (extends RecyclerView.Adapter<CatAdapter.CatViewHolder>)

```
public class MainActivity extends AppCompatActivity {
    private RecyclerView rView;
    private CatAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rView=findViewById(R.id.rview);
        adapter=new CatAdapter(this,getListCat());
        GridLayoutManager manager=new GridLayoutManager(this,3);
        rView.setLayoutManager(manager);
        rView.setAdapter(adapter);
    }
    private List<Cat> getListCat() {
        List<Cat> list=new ArrayList<>();
        list.add(new Cat(R.drawable.cat1,"Cat 1"));
        list.add(new Cat(R.drawable.cat2,"Cat 2"));
        list.add(new Cat(R.drawable.cat3,"Cat 3"));
        list.add(new Cat(R.drawable.cat4,"Cat 4"));
        list.add(new Cat(R.drawable.cat5,"Cat 5"));
        list.add(new Cat(R.drawable.cat6,"Cat 6"));
        return list;
    }
}
```

# Main_Activity.java

Example for RecycleView

Cat 1

Cat 2

Cat 3

Cat 4

Cat 5

Cat 6

# Events handling for RecycleView

```
holder.cardView.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {

Toast.makeText(mContext.getApplicationContext(),ca
t.getName(), Toast.LENGTH_SHORT).show();
    }
});
```

- Create Interface: CatItemListener.java
- Add private CatItemListener mCatItem; to file CatAdapter.java (add setter setClickListener(CatItemListener mCatItem))
- implements View.OnClickListener for CatViewHolder.java (v.setOnClickListener(this);)
- @Override

```
public void onClick(View view) {
        if(mCatItem!=null){
           mCatItem.onItemClick(view,getAdapterPosition());
        }
    }
```

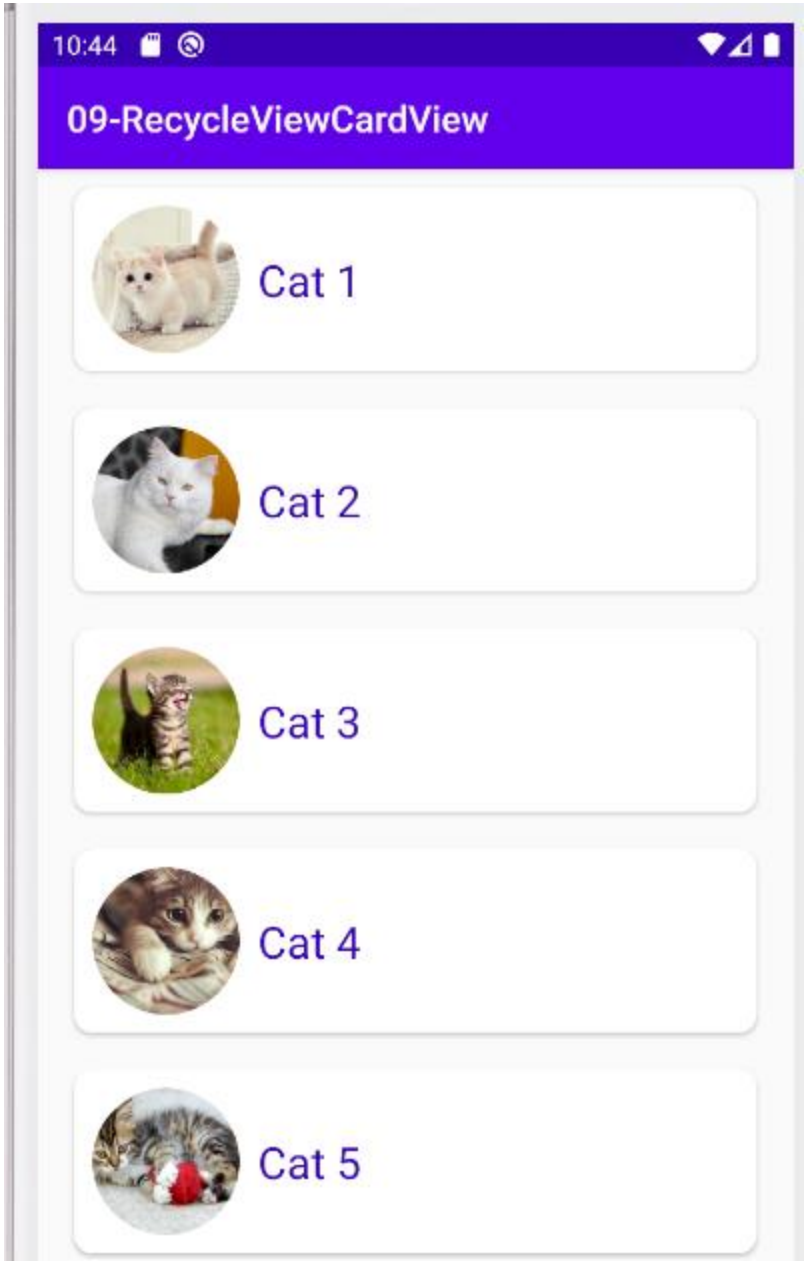
- MainActivity : implements CatAdapter.CatItemListener, and adapter.setClickListener(this);
- Method: onItemClick(View view, int position)

# CircleImageView

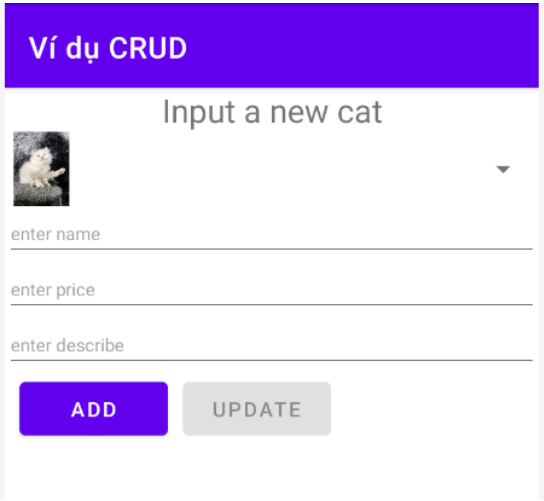
```
<de.hdodenhof.circleimageview.CircleImageView
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:id="@+id/img"
    android:src="@drawable/cat1"/>
```

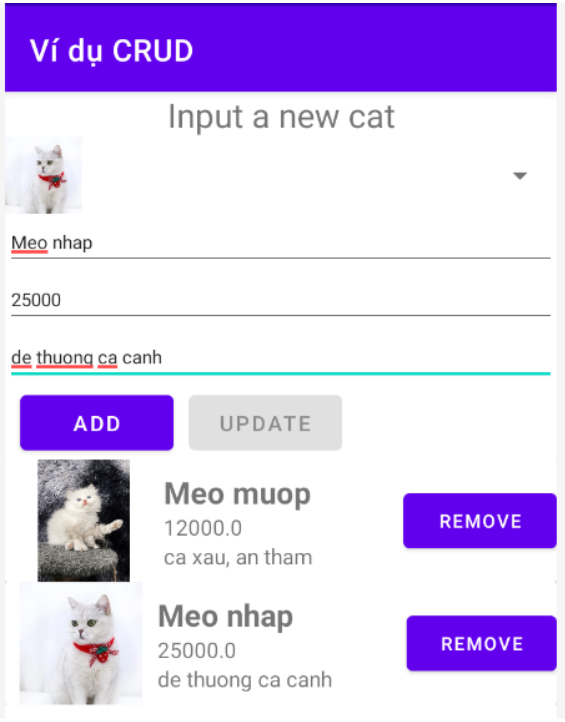
```
LinearLayoutManager manager=new LinearLayoutManager(this,
        RecyclerView.VERTICAL,false);
```

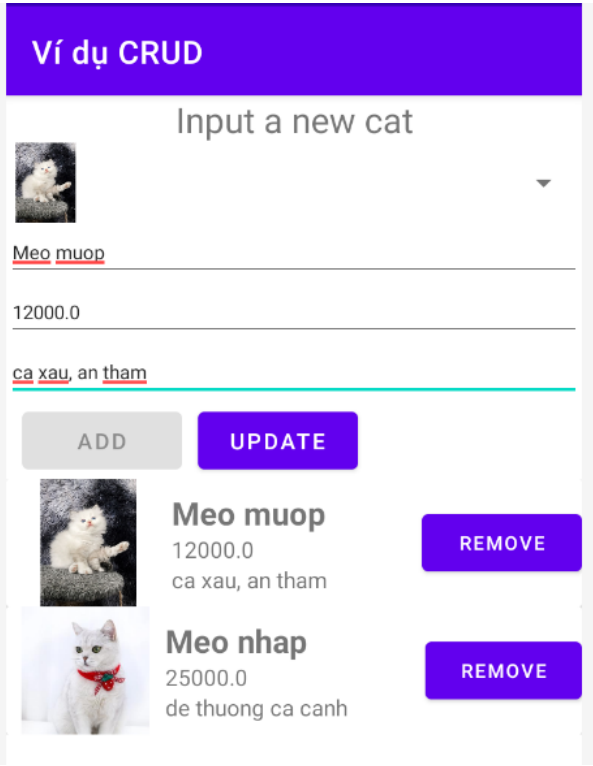
implementation
'de.hdodenhof:circleimageview:3.1.0'


10:44
09-RecycleViewCardView
Cat 1
Cat 2
Cat 3
Cat 4
Cat 5

# RecyclerView (09)



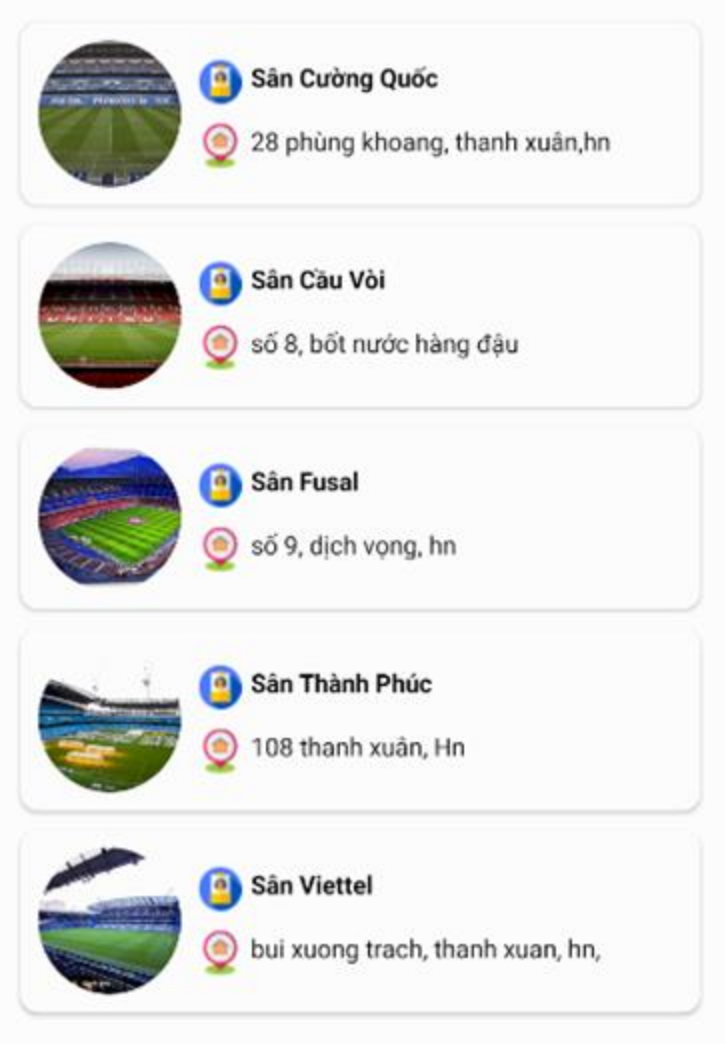



# Exercise 1


5554:avd17tes
4:41
Vidu_CustomLayout_ListView
Quản lý nhân viên
Mã NV:
Tên NV:
Giới tính:
Nữ
Nam
Nhập NV
Danh sách nhân viên:
ma1-Quach Tinh
ma2-Hoang Dung
ma3-Hong That Cong
ma4-Hoang Duoc su
ma5-Thanh Co

# Exercise 2




- input event listeners for card items

- End of Lesson 4

Thank you!