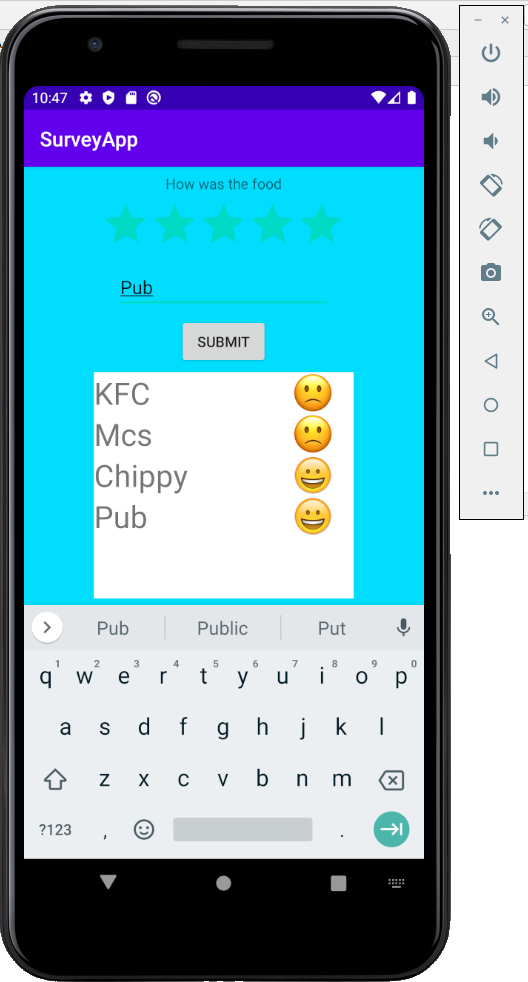
# A Survey Android App



### Aims:

To introduce developing an App for Android to handle survey data

### Objectives:

* Creating a user interface to interact with data
* Creating a model
* Managing a RecyclerView
* Creating an adapter between the model and the recycler view

Let’s review the lecture...

* How would you represent aggregations of objects with a class diagram?
* What is the ‘Collections API’?
* What is the advantage of an ArrayList over Arrays?
* What is the Adapter Pattern?
* How are Adapters implemented in Android java to display a list of objects in a list view?

## 1. Introduction

Here we’ll use Android Studio to develop an application able to handle multivariate data with a variety of controls.

## 2. Starting a new project

Let’s get started…

Open Android Studio and ‘Start a new Android Studio project’.

Choose an ‘Empty Activity’. Click [Next].

In the ‘Configure your new project’ dialog set your Application’s…

* name to ‘SurveyApp’ or similar
* package name to com.your\_initials.surveyapp
* ‘Save location’ to a suitable directory on your OneDrive
* ‘Language’ to **Java** **(NOT Kotlin!)**
* ‘Minimum API level’ to suggested default *…e.g. API 16: Android 4.1 (Jelly Bean)*

Click [Finish].

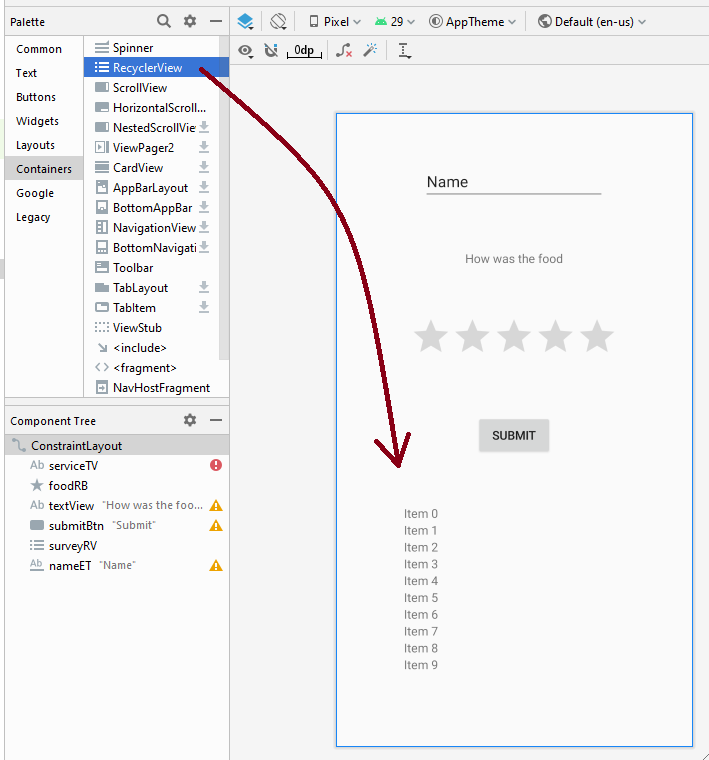
Allow a minute or so to let Gradle complete executing tasks. You will be presented with the Android Studio Integrated Development Environment.

## 3. Building the User Interface

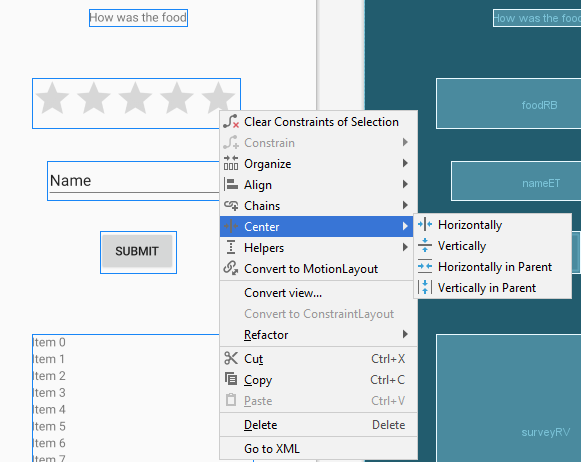
You have a default XML based resource file ‘activity\_main.xml’ where you create your user interface layout. The IDE provides a drag and drop style designer with which to specify user interface layout and contents.

Via the Editor or Android Explorer select **res/layout/activity\_main.xml** hence

* Click on and delete the ‘Hello World’ text view.
* Drag from the Palette onto your design a ‘Plain Text’, TextView, RatingBar, Button and RecyclerView.

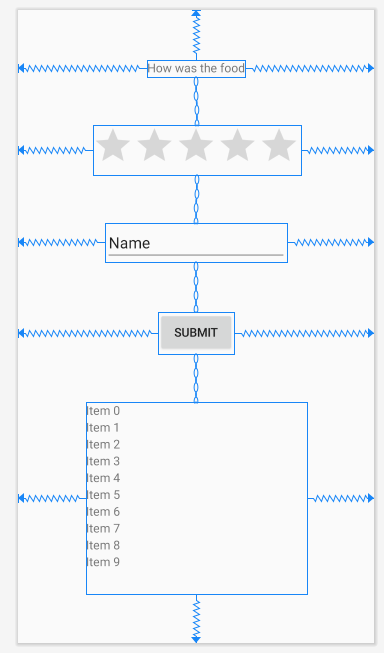


Drag mouse over to select all of these, hence right click to Center Horizontally and Vertically to apply constraints.



Via the Attributes pane for the …

* Plain Text set its ‘id’ to **nameET**
* TextView set its ‘text’ to “How was the food”
* RatingBar set its ‘id’ to **foodRB**
* Button set its ‘id’ to **submitBtn**, and its ‘text’ to “Submit”
* RecyclerView set its ‘id’ to **surveyRV**.



Run the application to test the look of the user interface.

## 4. Creating a model

In the **MainActivity** class declare **fields** which will be references to your widgets.

EditText **nameET**;  
RatingBar **foodRB**;  
Button **submitBtn**;  
RecyclerView **surveyRV**;

In the **onCreate()** method, bind these references with their respective XML based user interface components with ...

**nameET** = findViewById(R.id.***nameET***);  
**foodRB** = findViewById(R.id.***foodRB***);  
**submitBtn** = findViewById(R.id.***submitBtn***);  
**surveyRV** = findViewById(R.id.***surveyRV***);

We’ll want to manage data input as a list of objects. From **File->New->Java Class** create a new class called **SurveyItem**.

**public class** SurveyItem {  
 **private** String **name**;  
 **private float foodRating**;  
}

Use **Code->Generate** to generate a constructor and some getter methods, to give…

**public class** SurveyItem {  
 **private** String **name**;  
 **private float foodRating**;  
  
 **public** SurveyItem(String name, **float** foodRating) {  
 **this**.**name** = name;  
 **this**.**foodRating** = foodRating;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public float** getFoodRating() {  
 **return foodRating**;  
 }  
}

Hence in the **MainActivity** class declare as a field an array list of SurveryItem references.

ArrayList<SurveyItem> **survey**=**new** ArrayList<SurveyItem>();

From the **onCreate()** method populate this list with dummy data items.

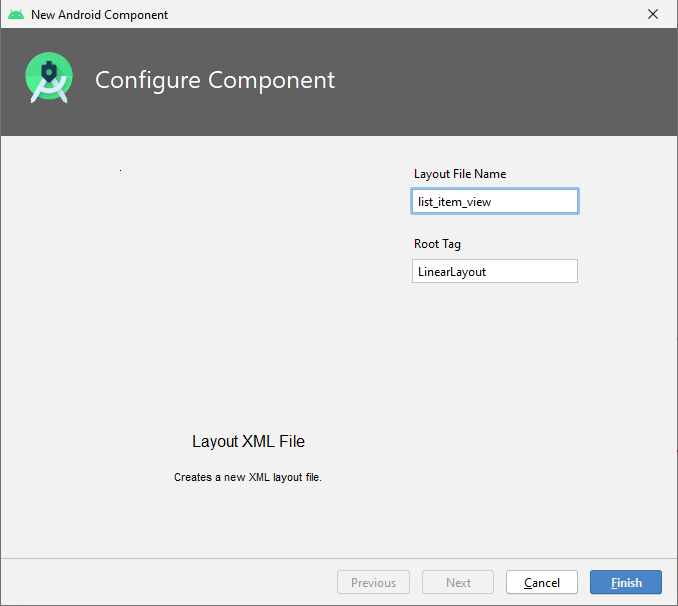
**survey**.add(**new** SurveyItem(**"KFC"**,1));  
**survey**.add(**new** SurveyItem(**"Mcs"**,2));  
**survey**.add(**new** SurveyItem(**"Chip Shop"**,4));

#### 5. Custom layout to represent item in Recycler view

Now that you have a model holding some data, next you’ll want users of your app to be able to interact with this model via the user interface.

You can create **custom layout** in a separate XML file that’ll be used by each list item, letting you specify text, etc to represent respective survey items. From the Package Explorer right click **res/layout** to create a **New -> XML->Layout XML** file.

From the resulting dialog box name the File: **list\_item\_view**.

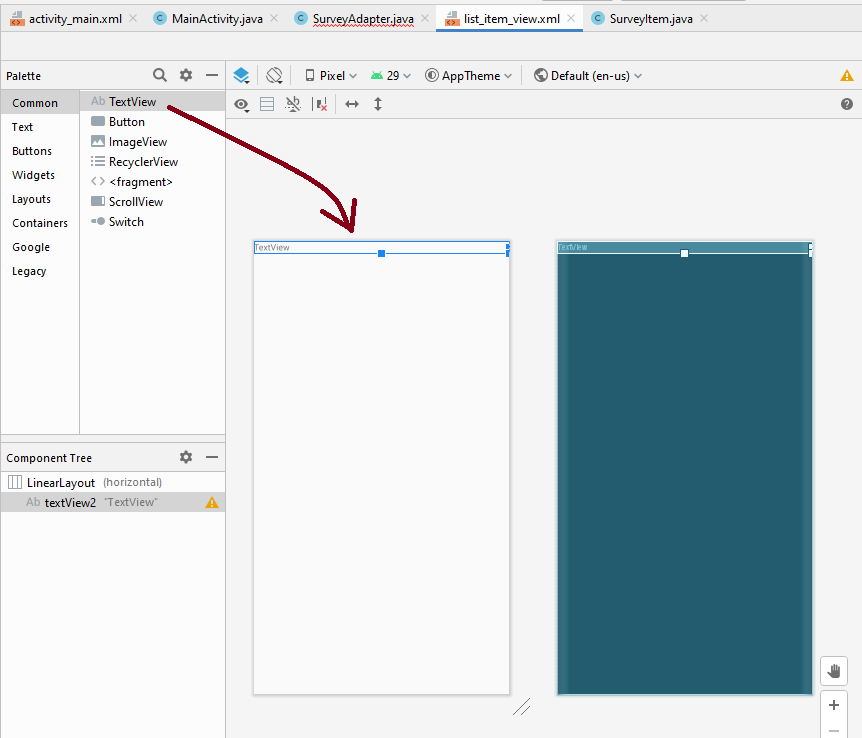


This’ll create a new custom layout you can drag widgets onto.

Open the Code view to see the **list\_item\_view.xml** file and set the layout\_height to wrap content.

**android:layout\_height="wrap\_content"**

Switch to the Design view. For now, just drag a TextView onto **list\_item\_view.xml**.

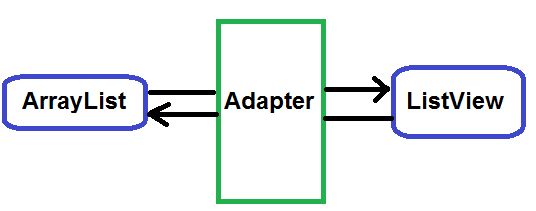


From the TextView’s attributes pane set its ‘id’ to **text\_in\_list\_item**

Be sure to save the XML every time you change an id.

## 6. Using an Adapter to display data in a list view

An adapter can be a bridge between data items and the widget used to display and interact with those data items. For example, an Adapter class can link items in your ArrayList with View widgets that can be displayed in a Recycler view.



From **File->New->Java Class** create a new class called **SurveyAdapter**.

Extend the class so it is a subclass of **RecyclerView.Adapter<SurveyAdapter.ViewHolder>**

Add the following field variables for your **SurveyAdapter** class

**private** ArrayList<SurveyItem> **survey**;  
**private int listItemLayout**;

Hence use **Code->Generate** to add a constructor and hover the mouse over the red underlined code to implement unimplemented methods. To give…

**public class** SurveyAdapter **extends** RecyclerView.Adapter<SurveyAdapter.ViewHolder> {  
 **private** ArrayList<SurveyItem> **survey**;  
 **private int listItemLayout**;  
  
 **public** SurveyAdapter(ArrayList<SurveyItem> survey, **int** listItemLayout) {  
 **this**.**survey** = survey;  
 **this**.**listItemLayout** = listItemLayout;  
 }  
  
 @NonNull  
 @Override  
 **public** SurveyAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,**int** viewType)

{  
 **return null**;  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull SurveyAdapter.ViewHolder holder, **int** position) {  
  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return** 0;  
 }

}

Click within the **SurveyAdapter** class to create a new inner class that extends **RecyclerView.ViewHolder** class and implements the **View.OnClickListener** interface. Hence let Android Student implement the constructor and unimplemented onClick() method.

**public class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener {  
 **public** ViewHolder(@NonNull View itemView) {  
 **super**(itemView);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
  
 }  
}

The full code for the **SurveyAdapter** class should look like…

**public class** SurveyAdapter **extends** RecyclerView.Adapter<SurveyAdapter.ViewHolder> {  
 **private** ArrayList<SurveyItem> **survey**;  
 **private int listItemLayout**;  
  
 **public** SurveyAdapter(ArrayList<SurveyItem> survey, **int** listItemLayout) {  
 **this**.**survey** = survey;  
 **this**.**listItemLayout** = listItemLayout;  
 }  
  
 @NonNull  
 @Override  
 **public** SurveyAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 **return null**;  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull SurveyAdapter.ViewHolder holder, **int** position) {  
  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return** 0;  
 }  
  
 **public class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener {  
 **public** ViewHolder(@NonNull View itemView) {  
 **super**(itemView);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
  
 }  
 }}

In the **ViewHolder** class add the field variable declaration…

**public** TextView **text\_in\_list\_item**;

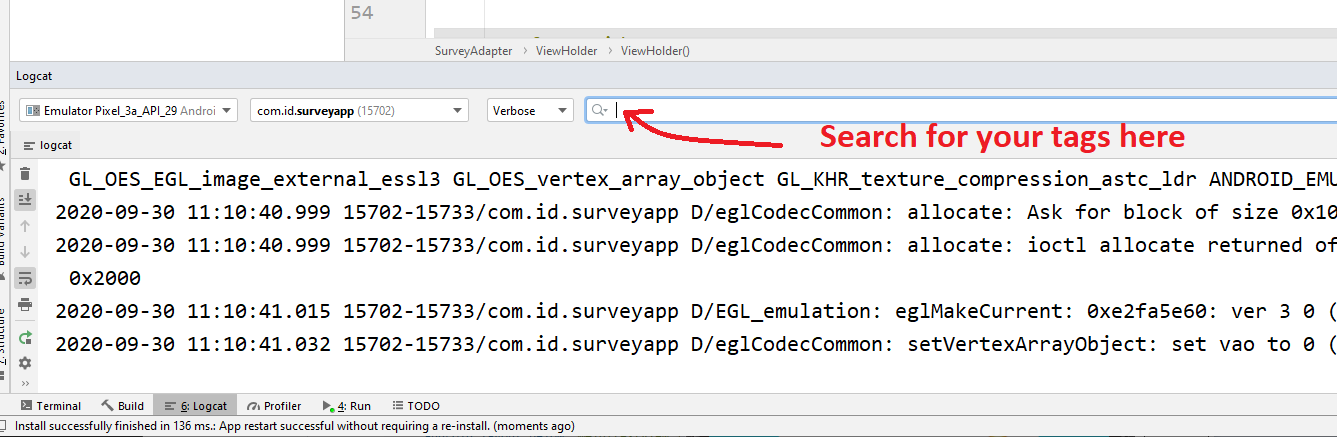
And in the **ViewHolder** constructor add…

**text\_in\_list\_item** = itemView.findViewById(R.id.***text\_in\_list\_item***);  
itemView.setOnClickListener(**this**);

In the onClick() and some code just to output a message on LogCat pane.

Log.*d*(**"MyTag"**, **"onClick "** + getLayoutPosition() + **" "** + **text\_in\_list\_item**.getText());

*LogCat is a great way to debug your code, its generates a stream of diagnostic messages when your app is pushed to the phone or emulator and during its running. You add your own messages with Log.d() and use the tag (e.g. MyTag) to search for the messages you create.*



In the onCreateViewHolder() method replace the *return null* with…

View view = LayoutInflater.*from*(parent.getContext()).inflate(**listItemLayout**, parent, **false**);  
ViewHolder myViewHolder = **new** ViewHolder(view);  
**return** myViewHolder;

This code uses a layout inflator which instantiates the **list\_item\_view.xml** file into its corresponding View object. The **ViewHolder** class is the inner class you defined as a wrapper for this object, to give it interaction.

In the **onBindViewHolder()** method add…

holder.**text\_in\_list\_item**.setText(**survey**.get(position).getName());

This method is used to update the contents of the Recycler view items shown when the survey array list changes.

Finally in the getItemCount() methods replace the *return 0* with …

**return survey**.size();

Add to the MainActivity class the field

SurveyAdapter **adapter**;

Hence to its onCreate() method…

**adapter** = **new** SurveyAdapter(**survey**,R.layout.***list\_item\_view***);  
 **surveyRV**.setLayoutManager(**new** LinearLayoutManager(**this**));  
 **surveyRV**.setItemAnimator(**new** DefaultItemAnimator());  
 **surveyRV**.setAdapter(**adapter**);

Which creates the bindings between the survey, adapter and recycler view.

Save and run the app, what does it do?

## EXERCISE

Ideally, we’d like user of the app to be able to add and rate more restaurants. Add the code to do this.

Note…

* in the **MainActivity** class’s code call the button’s **setOnClickListener()** method with an instance of an anonymous inner class which implements the **OnClickListener** interface.
* hence in the onClick() method you can add a new item to the survey with, for example…

**survey**.add(**new** SurveyItem(**nameET**.getText().toString(),**foodRB**.getRating()));

* After which **you also need to notify the adapter** of the change with…

**adapter**.notifyDataSetChanged();

#### EXERCISE

You might want to try adding other widgets to your **list\_item\_view.xml**

For example, you could drag onto **list\_item\_view.xml** an ‘ImageView’ that you set based on the food rating of a survey item, e.g. a happy or frowning emoji. You could add into your adapter’s **onBindViewHolder()** method the code snippet given below …

**if**(**survey**.get(position).getFoodRating()>3)  
 holder.**img\_in\_list\_item**.setImageResource(R.drawable.***smile***);  
**else** holder.**img\_in\_list\_item**.setImageResource(R.drawable.***frown***);

Where smile and frown are imported images, and **img\_in\_list\_item** is a field you have added to ViewHodler and bound with the imageView via findViewById().

#### EXERCISE

Explore interaction with the recycler view via the onClick() method, for example, letting the user select an element in the recycler view and enabling the user to edit it.

Expand the information which can be added to and held within a survey item, hence add other kinds of widgets to your **list\_item\_view.xml.**

Complete Code Example

MainActivity.java

**package** com.id.surveyapp;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.recyclerview.widget.DefaultItemAnimator;  
**import** androidx.recyclerview.widget.LinearLayoutManager;  
**import** androidx.recyclerview.widget.RecyclerView;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.RatingBar;  
**import** java.util.ArrayList;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 EditText **nameET**;  
 RatingBar **foodRB**;  
 Button **submitBtn**;  
 RecyclerView **surveyRV**;  
  
 ArrayList<SurveyItem> **survey**=**new** ArrayList<SurveyItem>();  
 SurveyAdapter **adapter**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **nameET** = findViewById(R.id.***nameET***);  
 **foodRB** = findViewById(R.id.***foodRB***);  
 **submitBtn** = findViewById(R.id.***submitBtn***);  
 **surveyRV** = findViewById(R.id.***surveyRV***);  
  
 **survey**.add(**new** SurveyItem(**"KFC"**,1));  
 **survey**.add(**new** SurveyItem(**"Mcs"**,2));  
 **survey**.add(**new** SurveyItem(**"Chippy"**,4));  
  
 **adapter** = **new** SurveyAdapter(**survey**,R.layout.***list\_item\_view***);  
 **surveyRV**.setLayoutManager(**new** LinearLayoutManager(**this**));  
 **surveyRV**.setItemAnimator(**new** DefaultItemAnimator());  
 **surveyRV**.setAdapter(**adapter**);  
  
 **submitBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **survey**.add(**new** SurveyItem(**nameET**.getText().toString(),**foodRB**.getRating()));  
 **adapter**.notifyDataSetChanged();  
 }  
 });  
 }  
}

SurveyItem.java

**package** com.id.surveyapp;  
  
**public class** SurveyItem {  
 **private** String **name**;  
 **private float foodRating**;  
  
 **public** SurveyItem(String name, **float** foodRating) {  
 **this**.**name** = name;  
 **this**.**foodRating** = foodRating;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public float** getFoodRating() {  
 **return foodRating**;  
 }  
}

SurveyAdapter.java

**package** com.id.surveyapp;  
  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
**import** java.util.ArrayList;  
  
**public class** SurveyAdapter **extends** RecyclerView.Adapter<SurveyAdapter.ViewHolder> {  
 **private** ArrayList<SurveyItem> **survey**;  
 **private int listItemLayout**;  
  
 **public** SurveyAdapter(ArrayList<SurveyItem> survey, **int** listItemLayout) {  
 **this**.**survey** = survey;  
 **this**.**listItemLayout** = listItemLayout;  
 }  
  
 @NonNull  
 @Override  
 **public** SurveyAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(**listItemLayout**, parent, **false**);  
 ViewHolder myViewHolder = **new** ViewHolder(view);  
 **return** myViewHolder;  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull SurveyAdapter.ViewHolder holder, **int** position) {  
 holder.**text\_in\_list\_item**.setText(**survey**.get(position).getName());  
 **if**(**survey**.get(position).getFoodRating()>3)  
 holder.**img\_in\_list\_item**.setImageResource(R.drawable.***smile***);  
 **else** holder.**img\_in\_list\_item**.setImageResource(R.drawable.***frown***);  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return survey**.size();  
 }  
  
 **public class** ViewHolder **extends** RecyclerView.ViewHolder **implements** View.OnClickListener {  
 **public** TextView **text\_in\_list\_item**;  
 **public** ImageView **img\_in\_list\_item**;  
  
 **public** ViewHolder(@NonNull View itemView) {  
 **super**(itemView);  
 **text\_in\_list\_item** = itemView.findViewById(R.id.***text\_in\_list\_item***);  
 **img\_in\_list\_item** = itemView.findViewById(R.id.***img\_in\_list\_item***);  
 itemView.setOnClickListener(**this**);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 Log.*d*(**"MyTag"**, **"onClick "** + getLayoutPosition() + **" "** + **text\_in\_list\_item**.getText());  
 }  
 }  
}

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"  
 android:background="@android:color/holo\_blue\_bright"  
 tools:context=".MainActivity"**>  
  
 <**RatingBar  
 android:id="@+id/foodRB"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toTopOf="@+id/nameET"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView"** />  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="How was the food"  
 app:layout\_constraintBottom\_toTopOf="@+id/foodRB"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**Button  
 android:id="@+id/submitBtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Submit"  
 app:layout\_constraintBottom\_toTopOf="@+id/surveyRV"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/nameET"** />  
  
 <**androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/surveyRV"  
 android:layout\_width="255dp"  
 android:layout\_height="222dp"  
 android:background="@android:color/background\_light"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/submitBtn"** />  
  
 <**EditText  
 android:id="@+id/nameET"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="Name"  
 app:layout\_constraintBottom\_toTopOf="@+id/submitBtn"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/foodRB"** />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

List\_item\_view.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"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**TextView  
 android:id="@+id/text\_in\_list\_item"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="TextView"  
 android:textSize="30sp"** />  
  
 <**ImageView  
 android:id="@+id/img\_in\_list\_item"  
 android:layout\_width="160dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 app:srcCompat="@drawable/frown"** />  
</**LinearLayout**>