

1/

The screenshot shows the Android Studio interface with the code editor and design view. The code editor displays the XML layout for a RecyclerView:

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

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/rv_Data"
        android:layout_marginStart="10dp"
        android:layout_marginBottom="10dp"/>

</LinearLayout>
```

The design view shows a purple header bar with the title "App Food List". Below it is a white list area containing ten items labeled "Item 0" through "Item 9".

The screenshot shows the Android Studio interface with the code editor and design view. The code editor displays the XML layout for a food item card:

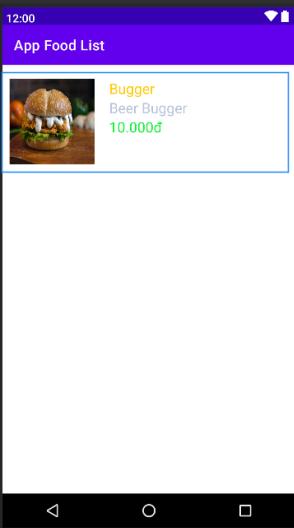
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout_item"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp">

    <ImageView
        android:id="@+id/imgFood"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:scaleType="fitXY"
        android:src="@drawable/burger_chicken" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_toEndOf="@+id/imgFood"
        android:layout_marginStart="20dp"
        android:orientation="vertical">
        <TextView
            android:id="@+id/food"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" ...>
```

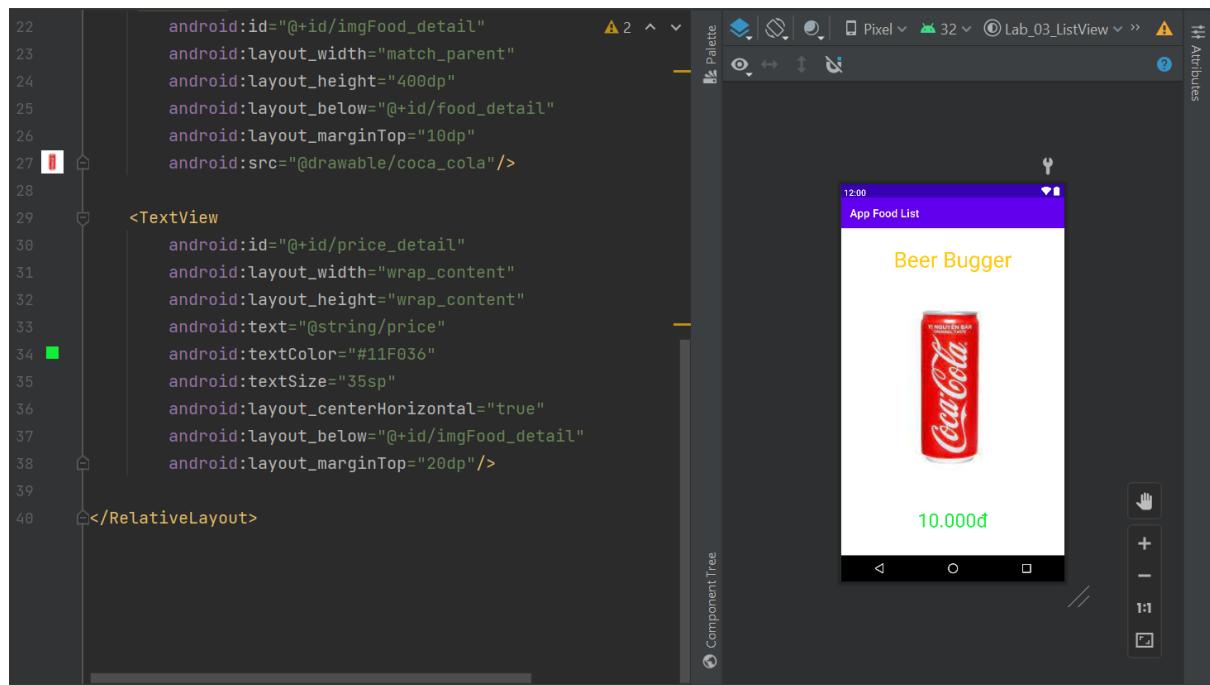
The design view shows a card layout for a food item. It features an image of a burger, the text "Bugger", "Beer Bugger", and "10.000d".

```
25
26
27     android:layout_height="wrap_content"      A 1 ^ v
28     android:text="Bugger"
29     android:textColor="#FFCA0D"
30     android:textSize="20sp"/>
31
32     <TextView
33         android:id="@+id/name"
34         android:layout_width="match_parent"
35         android:layout_height="wrap_content"
36         android:text="Beer Bugger"
37         android:textColor="#B8C4DC"
38         android:textSize="20sp"/>
39
40     <TextView
41         android:id="@+id/price"
42         android:layout_width="match_parent"
43         android:layout_height="wrap_content"
44         android:text="10.000đ"
45         android:textColor="#11F036"
46         android:textSize="20sp"/>
47
48 </LinearLayout>
49
50 </RelativeLayout>
```



```
1  <?xml version="1.0" encoding="utf-8"?>          A 2 ^ v
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".DetailActivity"
8      android:padding="10dp">
9
10     <TextView
11         android:id="@+id/food_detail"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:layout_alignParentTop="true"
15         android:layout_marginTop="20dp"
16         android:layout_centerHorizontal="true"
17         android:text="@string/name"
18         android:textColor="#FFCA0D"
19         android:textSize="40sp" />
20
21     <ImageView
22         android:id="@+id/imgFood_detail"
23         android:layout_width="match_parent"
24         android:layout_height="400dp"
```





The screenshot shows the Food.java code editor. The class definition is as follows:

```
4  public class Food implements Serializable {
5      private int image;
6      private String food;
7      private String name;
8      private String price;
9      public Food(int image, String food, String name, String price) {
10          this.image = image;
11          this.food = food;
12          this.name = name;
13          this.price = price;
14      }
15      public int getImage() { return image; }
16
17      public void setImage(int image) { this.image = image; }
18
19      public String getFood() { return food; }
20
21      public void setFood(String food) { this.food = food; }
22
23      public String getName() { return name; }
24
25      public void setName(String name) { this.name = name; }
26
27      public String getPrice() { return price; }
28
29      public void setPrice(String price) { this.price = price; }
30  }
```

```
C DetailActivity.java ×
3 import ...
9
10 public class DetailActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_detail);
16
17         Bundle bundle = getIntent().getExtras();
18         if (bundle == null){
19             return;
20         }
21
22         Food food = (Food) bundle.get("Food");
23
24         ImageView imgFoodDT = findViewById(R.id.imgFood_detail);
25         TextView txtPriceDT = findViewById(R.id.price_detail);
26         // Button btnBackList = findViewById(R.id.btnBackList);
27
28         TextView txtFoodDT = findViewById(R.id.food_detail);
29         txtFoodDT.setText(food.getName());
30         txtPriceDT.setText(food.getPrice());
31         imgFoodDT.setImageResource(food.getImage());
32     }
33 }
```

```
C FoodAdapter.java ×
3 import ...
16
17 public class FoodAdapter extends RecyclerView.Adapter<FoodAdapter.FoodViewHolder> {
18
19     private List<Food> listFood;
20     private IClickItemFoodListener iClickItemFoodListener;
21
22     public FoodAdapter(List<Food> listFood, IClickItemFoodListener listener) {
23         this.listFood = listFood;
24         this.iClickItemFoodListener = listener;
25     }
26
27     @NonNull
28     @Override
29     public FoodViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
30         View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_food, parent, attachToRoot: false
31         return new FoodViewHolder(view);
32     }
33
34     @Override
35     public void onBindViewHolder(@NonNull FoodViewHolder holder, int position) {
36         final Food food = listFood.get(position);
37         if (food == null){
38             return;
39         }
40         holder.imgFood.setImageResource(food.getImage());
41         holder.txtFood.setText(food.getFood());
```

```
C FoodAdapter.java ×
42         holder.txtName.setText(food.getName());
43         holder.txtPrice.setText(food.getPrice());
44
45         holder.layoutItem.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View view) { iClickItemFoodListener.OnClickItemFood(food); }
48         });
49     }
50 }
```

```
FoodAdapter.java
53     @Override
54     public int getItemCount() {
55         if (listFood != null){
56             return listFood.size();
57         }
58         return 0;
59     }
60
61     public class FoodViewHolder extends RecyclerView.ViewHolder{
62
63         private ImageView imgFood;
64         private TextView txtFood;
65         private TextView txtName;
66         private TextView txtPrice;
67         private RelativeLayout layoutItem;
68
69         public FoodViewHolder(@NonNull View itemView) {
70             super(itemView);
71             imgFood = itemView.findViewById(R.id.imgFood);
72             txtFood = itemView.findViewById(R.id.food);
73             txtName = itemView.findViewById(R.id.name);
74             txtPrice = itemView.findViewById(R.id.price);
75
76             layoutItem = itemView.findViewById(R.id.layout_item);
77         }
78     }
79 }
```

```
MainActivity.java
3 import ...
16
17     public class MainActivity extends AppCompatActivity {
18
19         private RecyclerView rcvData;
20         private FoodAdapter foodAdapter;
21
22
23         @Override
24         protected void onCreate(Bundle savedInstanceState) {
25             super.onCreate(savedInstanceState);
26             setContentView(R.layout.activity_main);
27
28             rcvData = findViewById(R.id.rcv_Data);
29             LinearLayoutManager linearLayoutManager = new LinearLayoutManager( context: this );
30             rcvData.setLayoutManager(linearLayoutManager);
31
32             RecyclerView.ItemDecoration itemDecoration = new DividerItemDecoration( context: this,DividerItemDecoration.
33             rcvData.addItemDecoration(itemDecoration);
34
35             foodAdapter = new FoodAdapter(getListFood(), new IClickItemFoodListener() {
36                 @Override
37                 public void OnClickItemFood(Food food) { onClickGoToDetail(food); }
38             });
39             rcvData.setAdapter(foodAdapter);
40
41         }
42     }
43 }
```

```
43
44     @
45         private List<Food> getListFood() {
46             List<Food> list = new ArrayList<>();
47             list.add(new Food(R.drawable.burger_1, food: "Burger", name: "Beet Bugger", price: "25.000d"));
48             list.add(new Food(R.drawable.pizza_cheese, food: "Pizza", name: "Cheese Pizza", price: "50.000d"));
49             list.add(new Food(R.drawable.burger_chicken, food: "Burger", name: "Chicken Bugger", price: "40.000d"));
50             list.add(new Food(R.drawable.chicken_fried_chilli, food: "Chicken Fried", name: "Fried Chicken Chilli", price: "35.000d"));
51             list.add(new Food(R.drawable.hot_dog, food: "Sandwich", name: "Hot Dog Sandwich", price: "45.000d"));
52             list.add(new Food(R.drawable.taco, food: "Taco", name: "Taco Shrimp", price: "60.000d"));
53             list.add(new Food(R.drawable.taco_beef, food: "Taco", name: "Taco Beef", price: "65.000d"));
54             list.add(new Food(R.drawable.coca_cola, food: "Coca", name: "Coca Cola Original", price: "15.000d"));
55             list.add(new Food(R.drawable.coca_cola_zeso, food: "Coca", name: "Coca Cola Zero", price: "15.000d"));
56
57         return list;
58     }
59
60     private void onClickGoToDetail (Food food) {
61         Intent intent = new Intent( packageContext: this, DetailActivity.class);
62         Bundle bundle = new Bundle();
63         bundle.putSerializable("Food", food);
64         bundle.putSerializable("Price", food);
65         intent.putExtras(bundle);
66         startActivity(intent);
67     }
68 }
```

```
1 <resources>
2     <string name="app_name">App Food List</string>
3     <string name="food">Bugger</string>
4     <string name="name">Beer Bugger</string>
5     <string name="price">10.000d</string>
6 </resources>
```

12:29

App Food List

Burger  
Beet Bugger  
25.000đ

Pizza  
Cheese Pizza  
50.000đ

Burger  
Chicken Bugger  
40.000đ

Chicken Fried  
Fried Chicken Chilli  
70.000đ

Sandwich

12:32

App Food List

Taco  
Taco Shrimp  
60.000đ

Taco  
Taco Beef  
65.000đ

Coca  
Coca Cola Original  
15.000đ

Coca  
Coca Cola Zero  
15.000đ

12:37

App Food List

## Fried Chicken Chilli



70.000đ

12:39

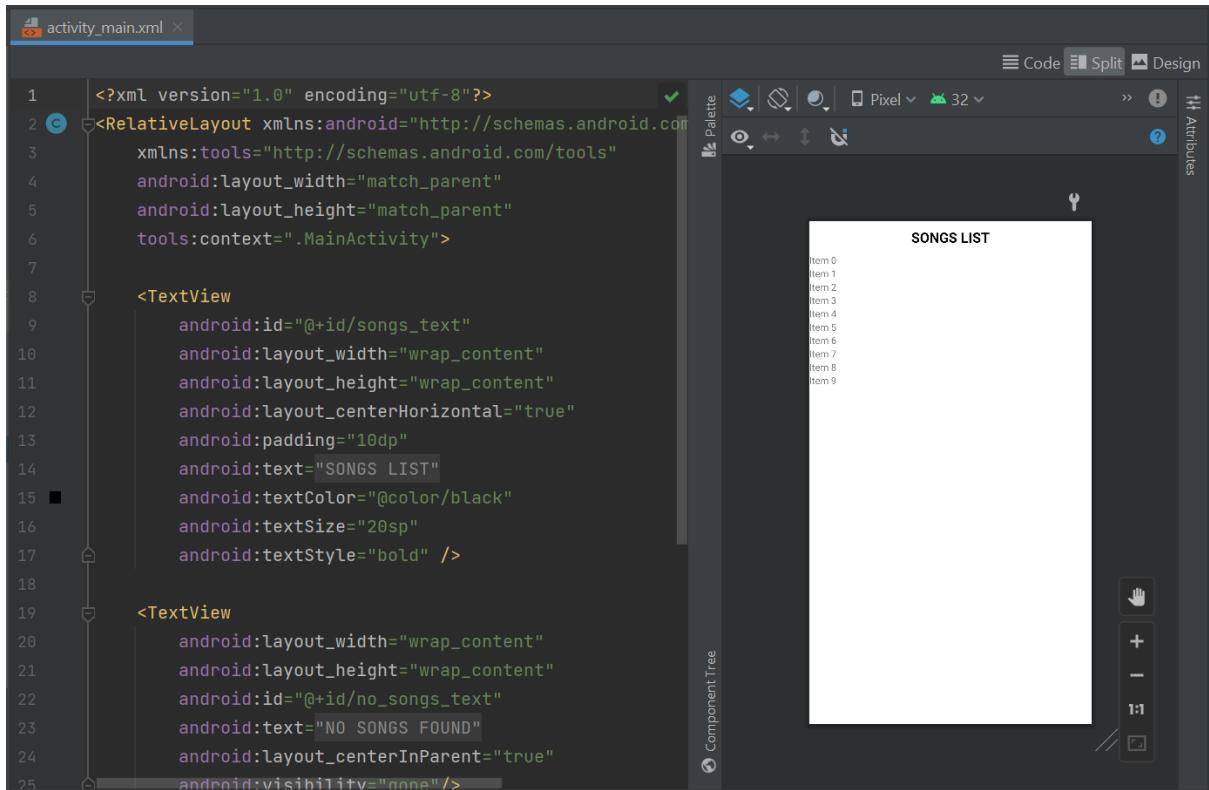
App Food List

## Coca Cola Zero



15.000đ

2/



activity\_main.xml

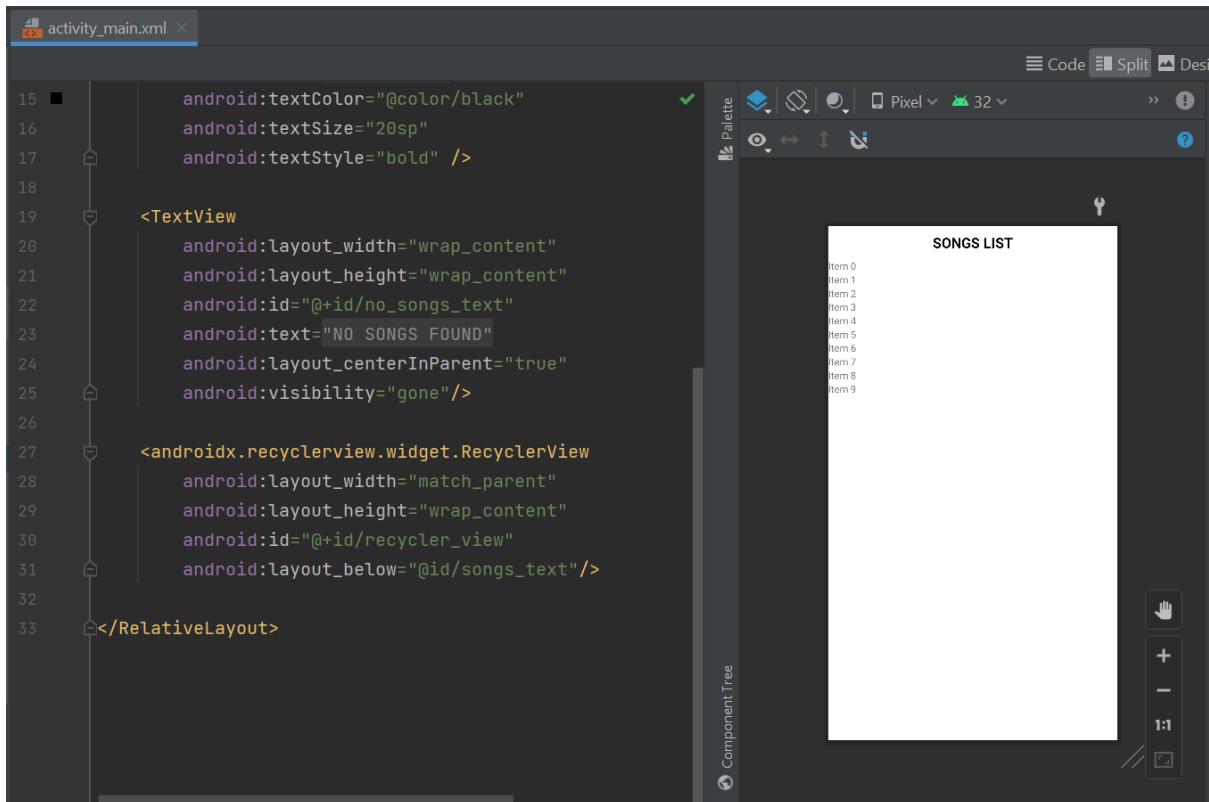
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/songs_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:padding="10dp"
        android:text="SONGS LIST"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/no_songs_text"
        android:text="NO SONGS FOUND"
        android:layout_centerInParent="true"
        android:visibility="gone" />

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/recycler_view"
        android:layout_below="@+id/songs_text" />

</RelativeLayout>
```



activity\_main.xml

```
15 <TextView
16     android:id="@+id/no_songs_text"
17     android:text="NO SONGS FOUND"
18     android:layout_centerInParent="true"
19     android:visibility="gone" />

20     <androidx.recyclerview.widget.RecyclerView
21         android:layout_width="match_parent"
22         android:layout_height="wrap_content"
23         android:id="@+id/recycler_view"
24         android:layout_below="@+id/no_songs_text" />

25 </RelativeLayout>
```

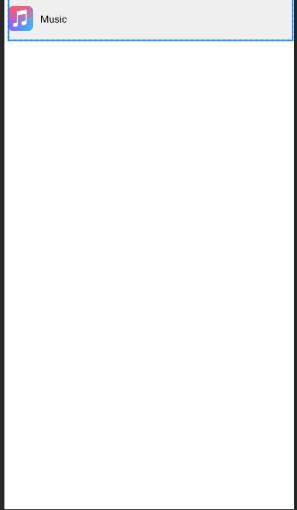
recycler\_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#EFEFEF"
    android:layout_marginTop="5dp"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    xmlns:tools="http://schemas.android.com/tools">

    <ImageView
        android:layout_width="36dp"
        android:layout_height="36dp"
        android:id="@+id/icon_view"
        android:layout_centerVertical="true"
        android:src="@drawable/music_icon"/>

    <TextView
        android:id="@+id/music_title_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:ellipsize="end"
        android:maxLines="1"
        android:padding="10dp"
        android:textColor="@color/black"
        tools:text="Music" />

```



recycler\_item.xml

```
android:layout_width="36dp"
android:layout_height="36dp"
android:id="@+id/icon_view"
android:layout_centerVertical="true"
android:src="@drawable/music_icon"/>

<TextView
    android:id="@+id/music_title_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_toEndOf="@+id/icon_view"
    android:ellipsize="end"
    android:maxLines="1"
    android:padding="10dp"
    android:textColor="@color/black"
    tools:text="Music" />

```



activity\_music\_player.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#33B2E8"
    tools:context=".MusicPlayerActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/song_title"
        android:text="Title of the song"
        android:singleLine="true"
        android:ellipsize="marquee"
        android:textColor="@color/white"
        android:textSize="30sp"
        android:layout_margin="20dp"
        android:padding="20dp"
    />

    <ImageView
        android:id="@+id/music_icon_big"
        android:layout_width="240dp"
        android:layout_height="240dp"
        android:src="@drawable/music_icon_big"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_above="@+id/controls"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_margin="20dp"
        android:padding="20dp"
    />

```

The screenshot shows the Android Studio interface with the 'Design' tab selected. The XML code defines a blue-themed music player screen. It features a large white treble clef icon in the center. Below it is a seek bar set to 0:00. At the bottom are three control buttons: back, play/pause, and forward. A text view above the icon displays the placeholder text 'Title of the song'.

activity\_music\_player.xml

```
    android:layout_height="240dp"
    android:layout_above="@+id/controls"
    android:layout_centerHorizontal="true"
    android:padding="20dp"
    android:src="@drawable/music_icon_big" />

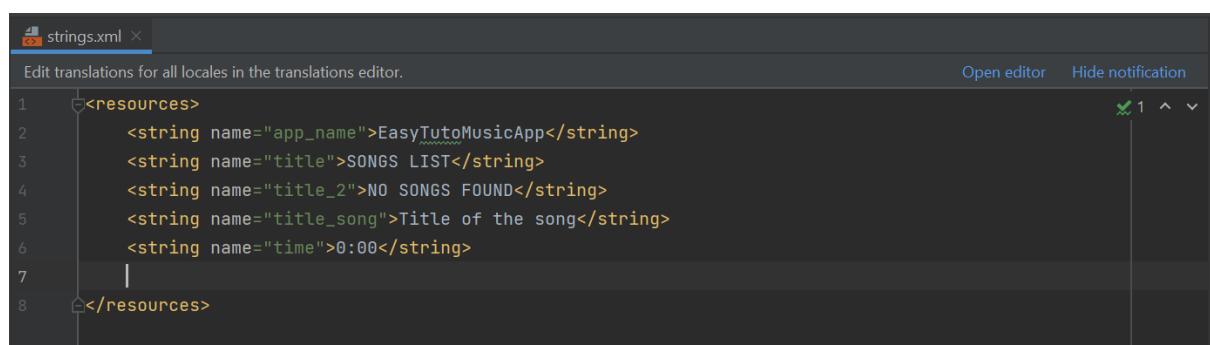
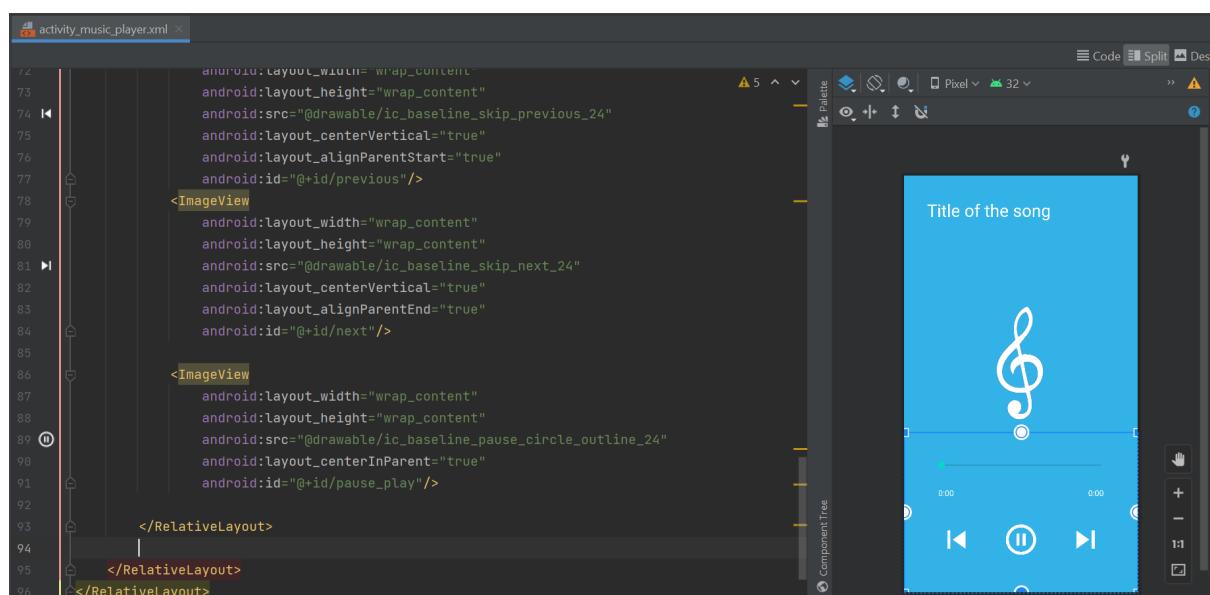
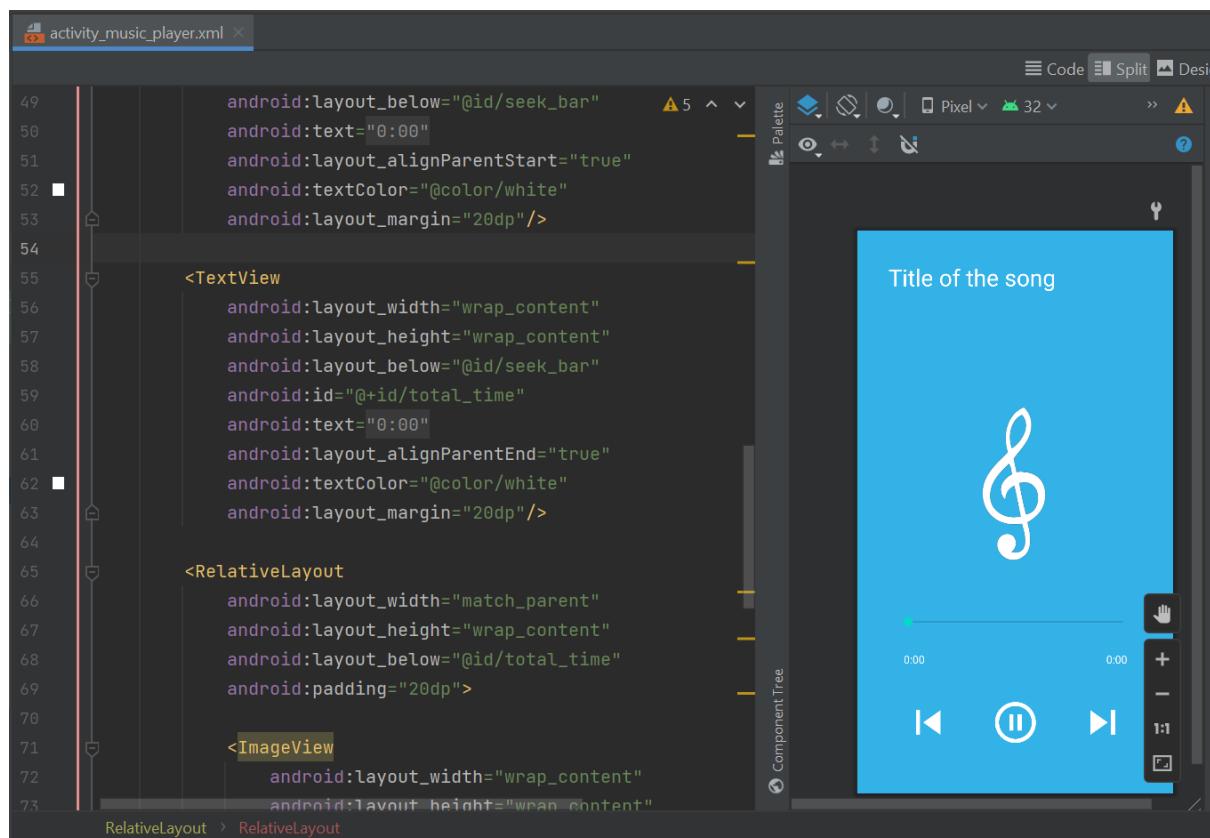
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:id="@+id/controls"
        android:padding="40dp">

        <SeekBar
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/seek_bar"
            android:layout_margin="10dp"
            android:backgroundTint="@color/white"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/current_time"
            android:layout_below="@+id/seek_bar"
            android:layout_centerHorizontal="true"/>
    

```

This screenshot shows the same XML code as the first one, but with additional components added to the 'controls' RelativeLayout. It includes a SeekBar and a TextView labeled 'current\_time' positioned below the seek bar. The overall layout remains consistent with the first screenshot, featuring a large central treble clef icon and control buttons at the bottom.



```
3 import java.io.Serializable;
4
5 public class AudioModel implements Serializable {
6     String path;
7     String title;
8     String duration;
9
10    public AudioModel(String path, String title, String duration) {
11        this.path = path;
12        this.title = title;
13        this.duration = duration;
14    }
15
16    public String getPath() { return path; }
17
18    public void setPath(String path) { this.path = path; }
19
20    public String getTitle() { return title; }
21
22    public void setTitle(String title) { this.title = title; }
23
24    public String getDuration() { return duration; }
25
26    public void setDuration(String duration) { this.duration = duration; }
27
28 }
29
30
31
32
33
34
35
36
37
38
39
40 }
```

```
3 import android.media.MediaPlayer;
4
5 public class MyMediaPlayer {
6     static MediaPlayer instance;
7
8     public static MediaPlayer getInstance(){
9         if(instance == null){
10             instance = new MediaPlayer();
11         }
12         return instance;
13     }
14
15     public static int currentIndex = -1;
16 }
17 }
```

```
MusicListAdapter.java
2 import ...
16
17     public class MusicListAdapter extends RecyclerView.Adapter<MusicListAdapter.ViewHolder>{
18         ArrayList<AudioModel> songsList;
19         Context context;
20         public MusicListAdapter(ArrayList<AudioModel> songsList, Context context) {
21             this.songsList = songsList;
22             this.context = context;
23         }
24
25         @Override
26         public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
27             View view = LayoutInflater.from(context).inflate(R.layout.recycler_item
28                 ,parent, attachToRoot: false);
29             return new MusicListAdapter.ViewHolder(view);
30         }
31
32         @Override
33         public void onBindViewHolder(MusicListAdapter.ViewHolder holder,
34                                     @SuppressLint("Recyclerview") int position) {
35             AudioModel songData = songsList.get(position);
36             holder.titleTextView.setText(songData.getTitle());
37
38             if(MyMediaPlayer.currentIndex==position){
39                 holder.titleTextView.setTextColor(Color.parseColor(colorString: "#FF0000"));
40             }else{
41                 holder.titleTextView.setTextColor(Color.parseColor(colorString: "#000000"));
42             }
43         }
44     }
45 }
```

```
MusicListAdapter.java
45     holder.itemView.setOnClickListener(new View.OnClickListener() {
46
47         @Override
48         public void onClick(View v) {
49             //navigate to another activity
50
51             MyMediaPlayer.getInstance().reset();
52             MyMediaPlayer.currentIndex = position;
53             Intent intent = new Intent(context, MusicPlayerActivity.class);
54             intent.putExtra(name: "LIST", songsList);
55             intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
56             context.startActivity(intent);
57         }
58     });
59 }
60 }
61
62         @Override
63         public int getItemCount() { return songsList.size(); }
64
65         public class ViewHolder extends RecyclerView.ViewHolder{
66
67             TextView titleTextView;
68             ImageView iconImageView;
69             public ViewHolder(View itemView) {
70                 super(itemView);
71                 titleTextView = itemView.findViewById(R.id.music_title_text);
72
73                 iconImageView = itemView.findViewById(R.id.icon_view);
74             }
75         }
76     }
77 }
```

```
MusicPlayerActivity.java
3 import ...
17
18 public class MusicPlayerActivity extends AppCompatActivity {
19
20     TextView titleTv,currentTimeTv,totalTimeTv;
21     SeekBar seekBar;
22     ImageView pausePlay,nextBtn,previousBtn,musicIcon;
23     ArrayList<AudioModel> songsList;
24     AudioModel currentSong;
25     MediaPlayer mediaPlayer = MyMediaPlayer.getInstance();
26     int x=0;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_music_player);
32
33         titleTv = findViewById(R.id.song_title);
34         currentTimeTv = findViewById(R.id.current_time);
35         totalTimeTv = findViewById(R.id.total_time);
36         seekBar = findViewById(R.id.seek_bar);
37         pausePlay = findViewById(R.id.pause_play);
38         nextBtn = findViewById(R.id.next);
39         previousBtn = findViewById(R.id.previous);
40         musicIcon = findViewById(R.id.music_icon_big);
41
42         titleTv.setSelected(true);

```

```
MusicPlayerActivity.java
43
44     songsList = (ArrayList<AudioModel>) getIntent().getSerializableExtra("LIST");
45
46     setResourcesWithMusic();
47
48     MusicPlayerActivity.this.runOnUiThread(new Runnable() {
49
50         @Override
51         public void run() {
52             if(mediaPlayer!=null){
53                 seekBar.setProgress(mediaPlayer.getCurrentPosition());
54                 currentTimeTv.setText(convertToMMSS(duration: mediaPlayer.getCurrentPosition()+""));
55
56                 if(mediaPlayer.isPlaying()){
57                     pausePlay.setImageResource(R.drawable.ic_baseline_pause_circle_outline_24);
58                     musicIcon.setRotation(x++);
59                 }else{
59                     pausePlay.setImageResource(R.drawable.ic_baseline_play_circle_outline_24);
60                     musicIcon.setRotation(0);
61                 }
62             }
63             new Handler().postDelayed( r: this, delayMillis: 100);
64         }
65     });
66
67     seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
68
69         @Override

```

```
MusicPlayerActivity.java
69
70     @Override
71     public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
72         if(mediaPlayer!=null && fromUser){
73             mediaPlayer.seekTo(progress);
74         }
75     }
76
77     @Override
78     public void onStartTrackingTouch(SeekBar seekBar) {
79     }
80
81     @Override
82     public void onStopTrackingTouch(SeekBar seekBar) {
83     }
84
85     });
86
87 }
88
89 void setResourcesWithMusic(){
90     currentSong = songsList.get(MyMediaPlayer.currentIndex);
91
92     titleTv.setText(currentSong.getTitle());
93
94     totalTimeTv.setText(convertToMMSS(currentSong.getDuration()));
95 }
```

```
MusicPlayerActivity.java
96
97     pausePlay.setOnClickListener(v-> pausePlay());
98     nextBtn.setOnClickListener(v-> playNextSong());
99     previousBtn.setOnClickListener(v-> playPreviousSong());
100
101     playMusic();
102 }
103
104 private void playMusic(){
105
106     mediaPlayer.reset();
107     try {
108         mediaPlayer.setDataSource(currentSong.getPath());
109         mediaPlayer.prepare();
110         mediaPlayer.start();
111         seekBar.setProgress(0);
112         seekBar.setMax(mediaPlayer.getDuration());
113     } catch (IOException e) {
114         e.printStackTrace();
115     }
116 }
117
118 private void playNextSong(){
119
120     if(MyMediaPlayer.currentIndex== songsList.size()-1)
121         return;
122     MyMediaPlayer.currentIndex +=1;
```

```
MusicPlayerActivity.java X
122     MyMediaPlayer.currentIndex +=1;
123     mediaPlayer.reset();
124     setResourcesWithMusic();
125
126 }
127
128 private void playPreviousSong(){
129     if(MyMediaPlayer.currentIndex== 0)
130         return;
131     MyMediaPlayer.currentIndex -=1;
132     mediaPlayer.reset();
133     setResourcesWithMusic();
134 }
135
136 private void pausePlay(){
137     if(mediaPlayer.isPlaying())
138         mediaPlayer.pause();
139     else
140         mediaPlayer.start();
141 }
142
143
144 @SuppressLint("DefaultLocale")
145 @ SuppressLint("DefaultLocale")
146 public static String convertToMMSS(String duration){
147     long millis = Long.parseLong(duration);
148     return String.format("%02d:%02d",
149             TimeUnit.MILLISECONDS.toMinutes(millis) % TimeUnit.HOURS.toMinutes( duration: 1),
150             TimeUnit.MILLISECONDS.toSeconds(millis) % TimeUnit.MINUTES.toSeconds( duration: 1));
151 }
```

```
 MainActivity.java ×
 3 import ...
20
21 public class MainActivity extends AppCompatActivity {
22
23     RecyclerView recyclerView;
24     TextView noMusicTextView;
25     ArrayList<AudioModel> songsList = new ArrayList<>();
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState) {
29         super.onCreate(savedInstanceState);
30         setContentView(R.layout.activity_main);
31
32         recyclerView = findViewById(R.id.recycler_view);
33         noMusicTextView = findViewById(R.id.no_songs_text);
34
35         if(checkPermission() == false){
36             requestPermission();
37             return;
38         }
39
40         String[] projection = {
41             MediaStore.Audio.Media.TITLE,
42             MediaStore.Audio.Media.DATA,
43             MediaStore.Audio.Media.DURATION
44     };

```

```
 MainActivity.java ×
 46     String selection = MediaStore.Audio.Media.IS_MUSIC + " != 0";
47
48     Cursor cursor = getContentResolver().query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
49         projection,selection,selectionArgs: null, sortOrder: null);
50     while(cursor.moveToNext()){
51         AudioModel songData = new AudioModel(cursor.getString( 1),
52             cursor.getString( 0),cursor.getString( 2));
53         if(new File(songData.getPath()).exists())
54             songsList.add(songData);
55     }
56
57     if(songsList.size()==0){
58         noMusicTextView.setVisibility(View.VISIBLE);
59     }else{
60         //recyclerview
61         recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
62         recyclerView.setAdapter(new MusicListAdapter(songsList,getApplicationContext()));
63     }
64
65 }
66
67 boolean checkPermission(){
68     int result = ContextCompat.checkSelfPermission( context: MainActivity.this,
69         Manifest.permission.READ_EXTERNAL_STORAGE);
70     if(result == PackageManager.PERMISSION_GRANTED){
71         return true;
72     }

```

```
MainActivity.java
70     if(result == PackageManager.PERMISSION_GRANTED){
71         return true;
72     }else{
73         return false;
74     }
75 }
76
77 void requestPermission(){
78     if(ActivityCompat.shouldShowRequestPermissionRationale(activity: MainActivity.this,
79             Manifest.permission.READ_EXTERNAL_STORAGE)){
80         Toast.makeText(context: MainActivity.this, text: "READ PERMISSION IS REQUIRED," +
81             "PLEASE ALLOW FROM SETTINGS",Toast.LENGTH_SHORT).show();
82     }else{
83         ActivityCompat.requestPermissions(activity: MainActivity.this,
84             new String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, requestCode: 123);
85     }
86 }
87
88 @Override
89 protected void onResume() {
90     super.onResume();
91     if(recyclerView!=null){
92         recyclerView.setAdapter(new MusicListAdapter(songsList,getApplicationContext()));
93     }
94 }
```

