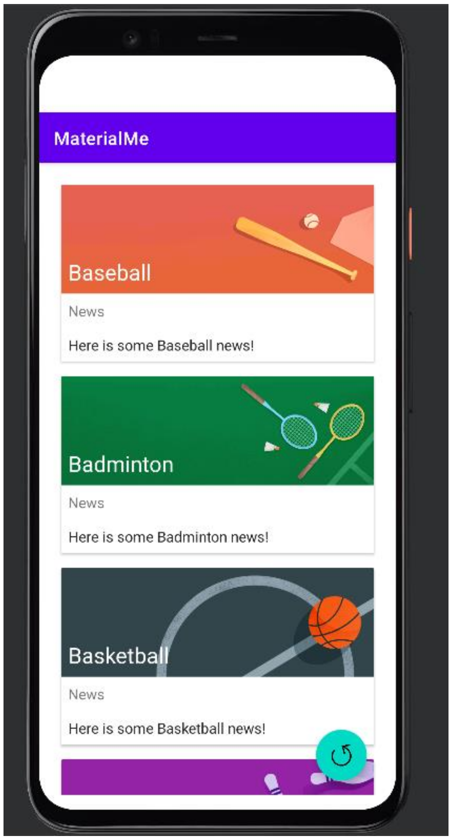
Mata Kuliah : Pemgraman Mobile – TI – S1

Pertemuan : Minggu 5

NIM : A11.2021.13509

Nama : Ainindzi Nur Meiza Pudjianto

1. App MaterialMe



MaterialMe

Baseball

News

Here is some Baseball news!

Badminton

News

Here is some Badminton news!

Basketball

News

Here is some Basketball news!

- **DetailActivity.java**

```
package com.example.materialme;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;

import com.bumptech.glide.Glide;

public class DetailActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);
// Initialize the views.
        TextView sportsTitle = findViewById(R.id.titleDetail);
        ImageView sportsImage = findViewById(R.id.sportsImageDetail);
// Set the text from the Intent extra.
        sportsTitle.setText(getIntent().getStringExtra("title"));
// Load the image using the Glide library and the Intent extra.

Glide.with(this).load(getIntent().getIntExtra("image_resource",0)).into(sport
sImage);
    }
}
```

- **MainActivity.java**

```
package com.example.materialme;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.ItemTouchHelper;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.res.TypedArray;
import android.os.Bundle;
import android.view.View;

import java.util.ArrayList;
import java.util.Collections;

public class MainActivity extends AppCompatActivity {
    // Member variables.
    private RecyclerView mRecyclerView;
    private ArrayList<Sport> mSportsData;
    private SportsAdapter mAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
// Initialize the RecyclerView.
        mRecyclerView = findViewById(R.id.recyclerView);

// Set the Layout Manager.
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
// Initialize the ArrayList that will contain the data.
        mSportsData = new ArrayList<>();
// Initialize the adapter and set it to the RecyclerView.
        mAdapter = new SportsAdapter(this, mSportsData);
        mRecyclerView.setAdapter(mAdapter);
// Get the data.
        initializeData();
// Helper class for creating swipe to dismiss and drag and drop
// functionality.
        ItemTouchHelper helper = new ItemTouchHelper(new
ItemTouchHelper.SimpleCallback(
                ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT |
                        ItemTouchHelper.DOWN | ItemTouchHelper.UP,
                ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT) {
            /**
             * Defines the drag and drop functionality.
             *
             * @param recyclerView The RecyclerView that contains the list
            items
             * @param viewHolder The SportsViewHolder that is being moved
             * @param target The SportsViewHolder that you are switching
            the
             * original one with.
             * @return true if the item was moved, false otherwise
             */
            @Override
            public boolean onMove(RecyclerView recyclerView,
                                  RecyclerView.ViewHolder viewHolder,
                                  RecyclerView.ViewHolder target) {

// Get the from and to positions.
                int from = viewHolder.getAdapterPosition();
                int to = target.getAdapterPosition();
// Swap the items and notify the adapter.
                Collections.swap(mSportsData, from, to);
                mAdapter.notifyItemMoved(from, to);
                return true;
            }
            /**
             * Defines the swipe to dismiss functionality.
             *
             * @param viewHolder The viewholder being swiped.
             * @param direction The direction it is swiped in.
             */
            @Override
            public void onSwiped(RecyclerView.ViewHolder viewHolder, int
direction) {
// Remove the item from the dataset.
                mSportsData.remove(viewHolder.getAdapterPosition());
// Notify the adapter.
                mAdapter.notifyItemRemoved(viewHolder.getAdapterPosition());
            }
        });
// Attach the helper to the RecyclerView.
        helper.attachToRecyclerView(mRecyclerView);
    }
    /**
     * Initialize the sports data from resources.

     */
    private void initializeData() {
// Get the resources from the XML file.
        String[] sportsList =
getResources().getStringArray(R.array.sports_titles);
        String[] sportsInfo =
getResources().getStringArray(R.array.sports_info);
        TypedArray sportsImageResources =
getResources().obtainTypedArray(R.array.sports_images);
// Clear the existing data (to avoid duplication).
        mSportsData.clear();
```

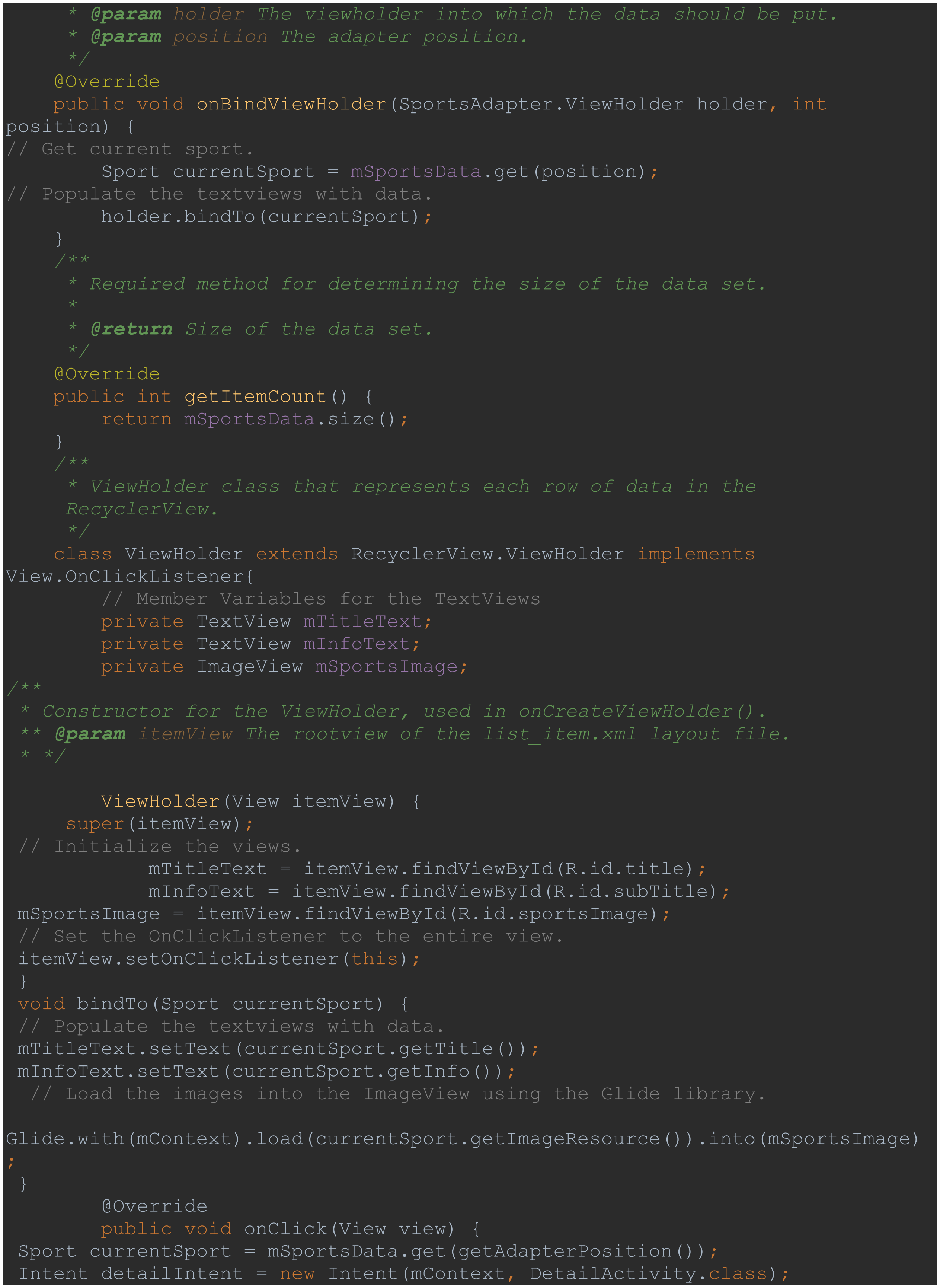
```
// Create the ArrayList of Sports objects with the titles and
// information about each sport
        for (int i = 0; i < sportsList.length; i++) {
            mSportsData.add(new Sport(sportsList[i], sportsInfo[i],
                    sportsImageResources.getResourceId(i, 0)));
        }
// Recycle the typed array.
        sportsImageResources.recycle();
// Notify the adapter of the change.
        mAdapter.notifyDataSetChanged();
    }
    /**
     * onClick method for th FAB that resets the data.
     *
     * @param view The button view that was clicked.
     */
    public void resetSports(View view) {
        initializeData();
    }
}
```

- **Sport.java**

```
package com.example.materialme;

class Sport {
// Member variables representing the title and information about the sport.
    private String title;
    private String info;
    private final int imageResource;
    /**
     * Constructor for the Sport data model.
     *
     * @param title The name if the sport.
     * @param info Information about the sport.
     */
    public Sport(String title, String info, int imageResource) {
        this.title = title;
        this.info = info;
        this.imageResource = imageResource;
    }
    /**
     * Gets the title of the sport.
     *
     * @return The title of the sport.
     */
    String getTitle() {
        return title;
    }
    /**
     * Gets the info about the sport.
     *
     * @return The info about the sport.
     */
    String getInfo() {
        return info;
    }
    public int getImageResource() {
        return imageResource;
    }
}
```

- **SportsAdapter.java**

```
package com.example.materialme;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.ArrayList;

class SportsAdapter extends RecyclerView.Adapter<SportsAdapter.ViewHolder>
{
    // Member variables.
    private ArrayList<Sport> mSportsData;
    private Context mContext;

    /**
     * Constructor that passes in the sports data and the context.
     *
     * @param sportsData ArrayList containing the sports data.
     * @param context Context of the application.
     */
    SportsAdapter(Context context, ArrayList<Sport> sportsData) {
        this.mSportsData = sportsData;
        this.mContext = context;
    }
    /**
     * Required method for creating the viewholder objects.
     *
     * @param parent The ViewGroup into which the new View will be added
     * after it is bound to an adapter position.
     * @param viewType The view type of the new View.
     * @return The newly created ViewHolder.
     */
    @Override
    public SportsAdapter.ViewHolder onCreateViewHolder(
            ViewGroup parent, int viewType) {
        return new ViewHolder(LayoutInflater.from(mContext).
                inflate(R.layout.list_item, parent, false));
    }
    /**
     * Required method that binds the data to the viewholder.
     *
```



```
     * @param holder The viewholder into which the data should be put.
     * @param position The adapter position.
     */
    @Override
    public void onBindViewHolder(SportsAdapter.ViewHolder holder, int
position) {
// Get current sport.
        Sport currentSport = mSportsData.get(position);
// Populate the textviews with data.
        holder.bindTo(currentSport);
    }
    /**
     * Required method for determining the size of the data set.
     *
     * @return Size of the data set.
     */
    @Override
    public int getItemCount() {
        return mSportsData.size();
    }
    /**
     * ViewHolder class that represents each row of data in the
     RecyclerView.
     */
    class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener{
        // Member Variables for the TextViews
        private TextView mTitleText;
        private TextView mInfoText;
        private ImageView mSportsImage;
/**
 * Constructor for the ViewHolder, used in onCreateViewHolder().
 ** @param itemView The rootview of the list_item.xml layout file.
 * */

        ViewHolder(View itemView) {
     super(itemView);
 // Initialize the views.
            mTitleText = itemView.findViewById(R.id.title);
            mInfoText = itemView.findViewById(R.id.subTitle);
 mSportsImage = itemView.findViewById(R.id.sportsImage);
 // Set the OnClickListener to the entire view.
 itemView.setOnClickListener(this);
 }
 void bindTo(Sport currentSport) {
 // Populate the textviews with data.
 mTitleText.setText(currentSport.getTitle());
 mInfoText.setText(currentSport.getInfo());
  // Load the images into the ImageView using the Glide library.

Glide.with(mContext).load(currentSport.getImageResource()).into(mSportsImage)
;
 }
        @Override
        public void onClick(View view) {
 Sport currentSport = mSportsData.get(getAdapterPosition());
 Intent detailIntent = new Intent(mContext, DetailActivity.class);
 detailIntent.putExtra("title", currentSport.getTitle());
 detailIntent.putExtra("image_resource", currentSport.getImageResource());
 mContext.startActivity(detailIntent);
 }
    }
}
```

- **Activity_detail.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <RelativeLayout xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:context="com.ryn.materialme.DetailActivity">
        <ImageView
            android:id="@+id/sportsImageDetail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:adjustViewBounds="true" />
        <TextView
            android:id="@+id/titleDetail"
            style="@style/TextAppearance.AppCompat.Headline"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBottom="@id/sportsImageDetail"
            android:padding="16dp"
            android:text="@string/title_placeholder"
            android:theme="@style/ThemeOverlay.AppCompat.Dark" />
        <TextView
            android:id="@+id/newsTitleDetail"
            style="@style/TextAppearance.AppCompat.Subhead"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/sportsImageDetail"
            android:padding="16dp"
            android:text="@string/news_placeholder"
            android:textColor="?android:textColorSecondary" />
        <TextView
            android:id="@+id/subTitleDetail"
            style="@style/TextAppearance.AppCompat.Subhead"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/newsTitleDetail"
            android:padding="16dp"
            android:text="@string/subtitle_detail_text" />
    </RelativeLayout>
</ScrollView>
```

- **Activity_main.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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.ryn.materialme.MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"

android:scrollbars="vertical"></androidx.recyclerview.widget.RecyclerView>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:minHeight="48dp"
        android:onClick="resetSports"
        android:src="@drawable/ic_reset"
        android:tint="@android:color/white"
        tools:ignore="ImageContrastCheck,SpeakableTextPresentCheck" />
</RelativeLayout>
```

- **List_item.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ImageView
            android:id="@+id/sportsImage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:adjustViewBounds="true"/>
        <TextView
            android:id="@+id/title"
            style="@style/TextAppearance.AppCompat.Headline"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="8dp"
            android:layout_alignBottom="@id/sportsImage"
            android:theme="@style/ThemeOverlay.AppCompat.Dark"
            android:text="@string/title_placeholder" />
        <TextView
            android:id="@+id/newsTitle"
            style="@style/TextAppearance.AppCompat.Subhead"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/sportsImage"
            android:textColor="?android:textColorSecondary"
            android:padding="8dp"
            android:text="@string/news_placeholder" />
        <TextView
            android:id="@+id/subTitle"
            style="@style/TextAppearance.AppCompat.Subhead"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/newsTitle"
            android:padding="8dp"
            android:text="@string/sports_info_placeholder" />
    </RelativeLayout>
</androidx.cardview.widget.CardView>
```

- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

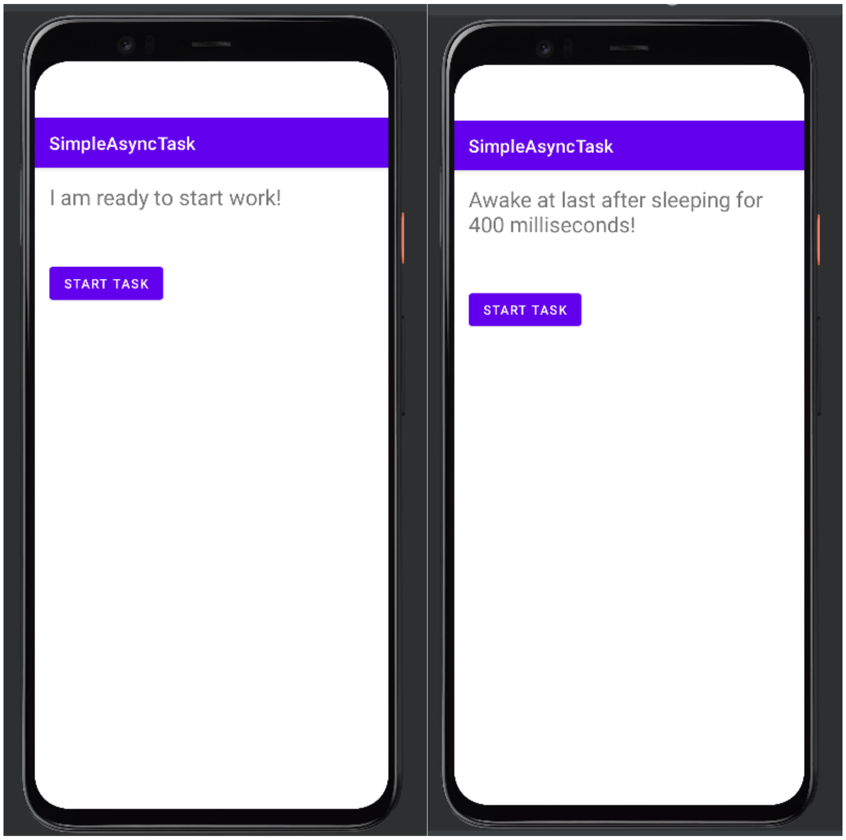
- **Dimens.xml**

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

- **Strings.xml**

```
<resources>
    <string name="app_name">MaterialMe</string>
    <string name="title_placeholder">Title</string>
    <string name="news_placeholder">News</string>
    <string name="sports_info_placeholder">Here is some news</string>
    <string name="subtitle_detail_text">Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Praesent ultrices congue rutrum.
Phasellus elementum ipsum ac convallis aliquam. Suspendisse
eleifend
eros a enim faucibus mollis. Nunc placerat, est vitae vestibulum
blandit, dolor diam fringilla tellus, eu euismod mauris neque
at neque.</string>
    <string-array name="sports_titles">
        <item>Baseball</item>
        <item>Badminton</item>
        <item>Basketball</item>
        <item>Bowling</item>
        <item>Cycling</item>
        <item>Golf</item>
        <item>Running</item>
        <item>Soccer</item>
        <item>Swimming</item>
        <item>Table Tennis</item>
        <item>Tennis</item>
    </string-array>
    <string-array name="sports_info">
        <item>Here is some Baseball news!</item>
        <item>Here is some Badminton news!</item>
        <item>Here is some Basketball news!</item>
        <item>Here is some Bowling news!</item>
        <item>Here is some Cycling news!</item>
        <item>Here is some Golf news!</item>
        <item>Here is some Running news!</item>
        <item>Here is some Soccer news!</item>
        <item>Here is some Swimming news!</item>
        <item>Here is some Table Tennis news!</item>
        <item>Here is some Tennis news!</item>
    </string-array>
    <array name="sports_images">
        <item>@drawable/img_baseball</item>
        <item>@drawable/img_badminton</item>
        <item>@drawable/img_basketball</item>
        <item>@drawable/img_bowling</item>
        <item>@drawable/img_cycling</item>
        <item>@drawable/img_golf</item>
        <item>@drawable/img_running</item>
        <item>@drawable/img_soccer</item>
        <item>@drawable/img_swimming</item>
        <item>@drawable/img_tabletennis</item>
        <item>@drawable/img_tennis</item>
    </array>
</resources>
```

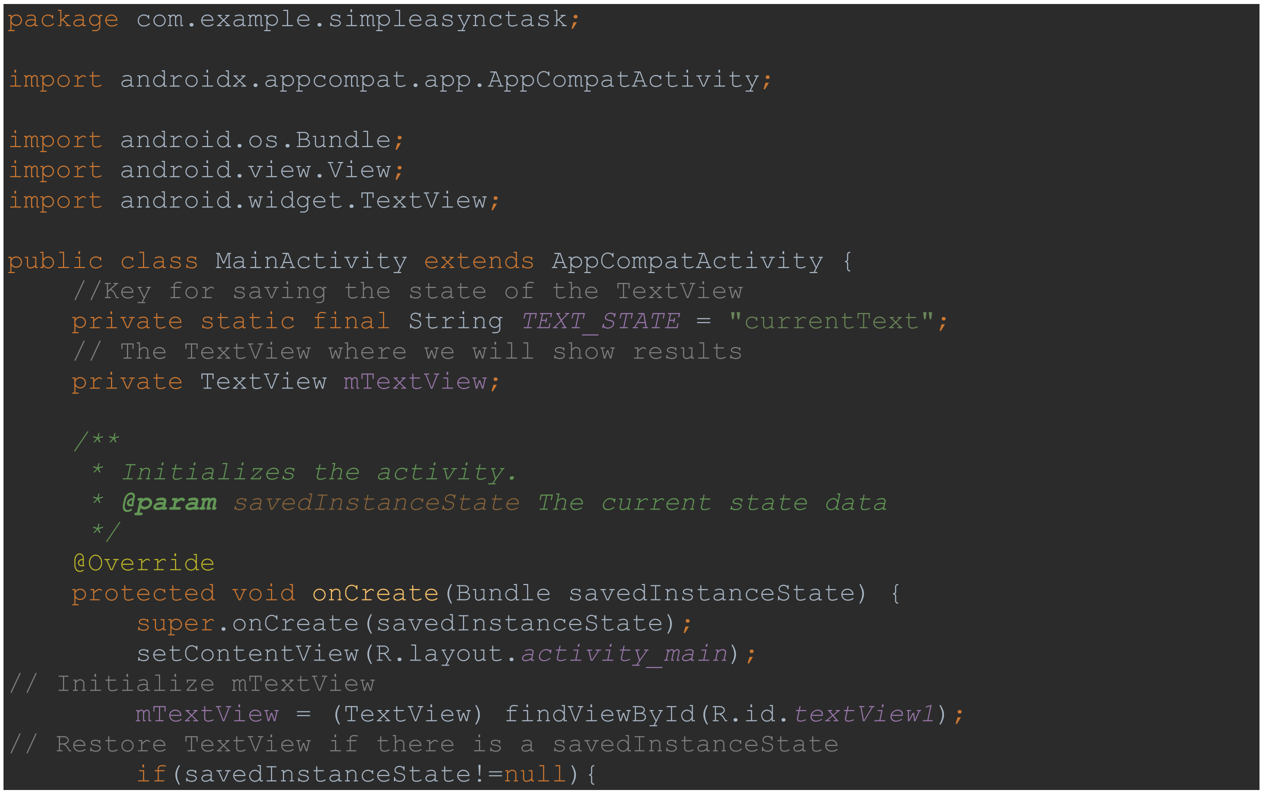
2. App SimpleAsyncTask



I am ready to start work!

Awake at last after sleeping for 400 milliseconds!

- **MainActivity.java**



```
package com.example.simpleasynctask;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    //Key for saving the state of the TextView
    private static final String TEXT_STATE = "currentText";
    // The TextView where we will show results
    private TextView mTextView;

    /**
     * Initializes the activity.
     * @param savedInstanceState The current state data
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
// Initialize mTextView
        mTextView = (TextView) findViewById(R.id.textView1);
// Restore TextView if there is a savedInstanceState
        if(savedInstanceState!=null){
            mTextView.setText(savedInstanceState.getString(TEXT_STATE));
        }
    }
    /**
     * Handles the onClick for the "Start Task" button. Launches the
     AsyncTask
     * which performs work off of the UI thread.
     * @param view The view (Button) that was clicked.
     */
    public void startTask (View view) {
// Put a message in the text view
        mTextView.setText(R.string.napping);
// Start the AsyncTask.
// The AsyncTask has a callback that will update the text view.
        new SimpleAsyncTask(mTextView).execute();
    }
    /**
     * Saves the contents of the TextView to restore on configuration
     change.
     * @param outState The bundle in which the state of the activity is
    saved when
     * it is spontaneously destroyed.
     */
    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
// Save the state of the TextView
        outState.putString(TEXT_STATE, mTextView.getText().toString());
    }
}
```

- **SimpleAsyncTask.java**

```
package com.example.simpleasynctask;

import android.os.AsyncTask;
import android.widget.TextView;

import java.util.Random;

public class SimpleAsyncTask extends AsyncTask<Void, Void, String> {
    // The TextView where we will show results
    private TextView mTextView;
// Constructor that provides a reference to the TextView from theMainActivity
    public SimpleAsyncTask(TextView tv) {
        mTextView = tv;
    }
    /**
     * Runs on the background thread.
     *

     * @param voids No parameters in this use case.
     * @return Returns the string including the amount of time that
     * the background thread slept.
     */
    @Override
    protected String doInBackground(Void... voids) {
// Generate a random number between 0 and 10
        Random r = new Random();
        int n = r.nextInt(11);
// Make the task take long enough that we have
// time to rotate the phone while it is running
        int s = n * 200;
// Sleep for the random amount of time
        try {
            Thread.sleep(s);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
// Return a String result
        return "Awake at last after sleeping for " + s + " milliseconds!";
    }

    /**
     * Does something with the result on the UI thread; in this case
     * updates the TextView.
     */
    protected void onPostExecute(String result) {
        mTextView.setText(result);
    }
}
```

- **activity_main.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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/ready_to_start"
        android:id = "@+id/textView1"
        android:textSize="24sp"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/start_task"
        android:id="@+id/button"
        android:layout_marginTop="56dp"
        android:onClick="startTask" />
</LinearLayout>
```

- **colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

- **dimens.xml**

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

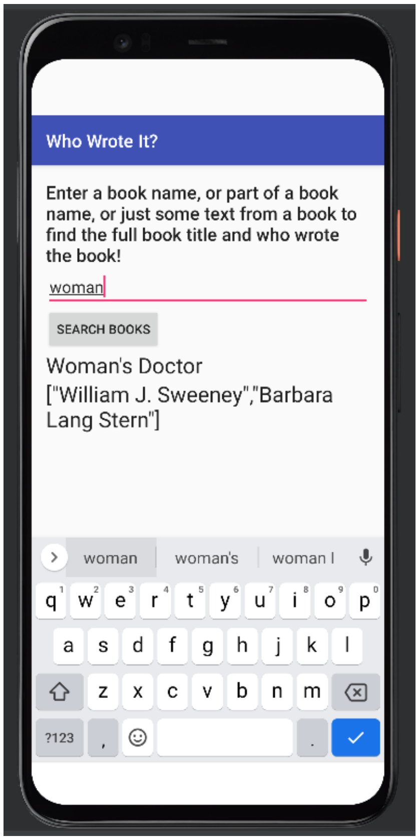
- **strings.xml**

```
<resources>
    <string name="app_name">SimpleAsyncTask</string>
    <string name="ready_to_start">I am ready to start work!</string>
    <string name="start_task">Start Task</string>
    <string name="napping">"Napping..."</string>
</resources>
```

- **themes.xml**

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.SimpleAsyncTask"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```

3. App WhoWroteIt



Who Wrote It?

Enter a book name, or part of a book name, or just some text from a book to find the full book title and who wrote the book!

woman

SEARCH BOOKS

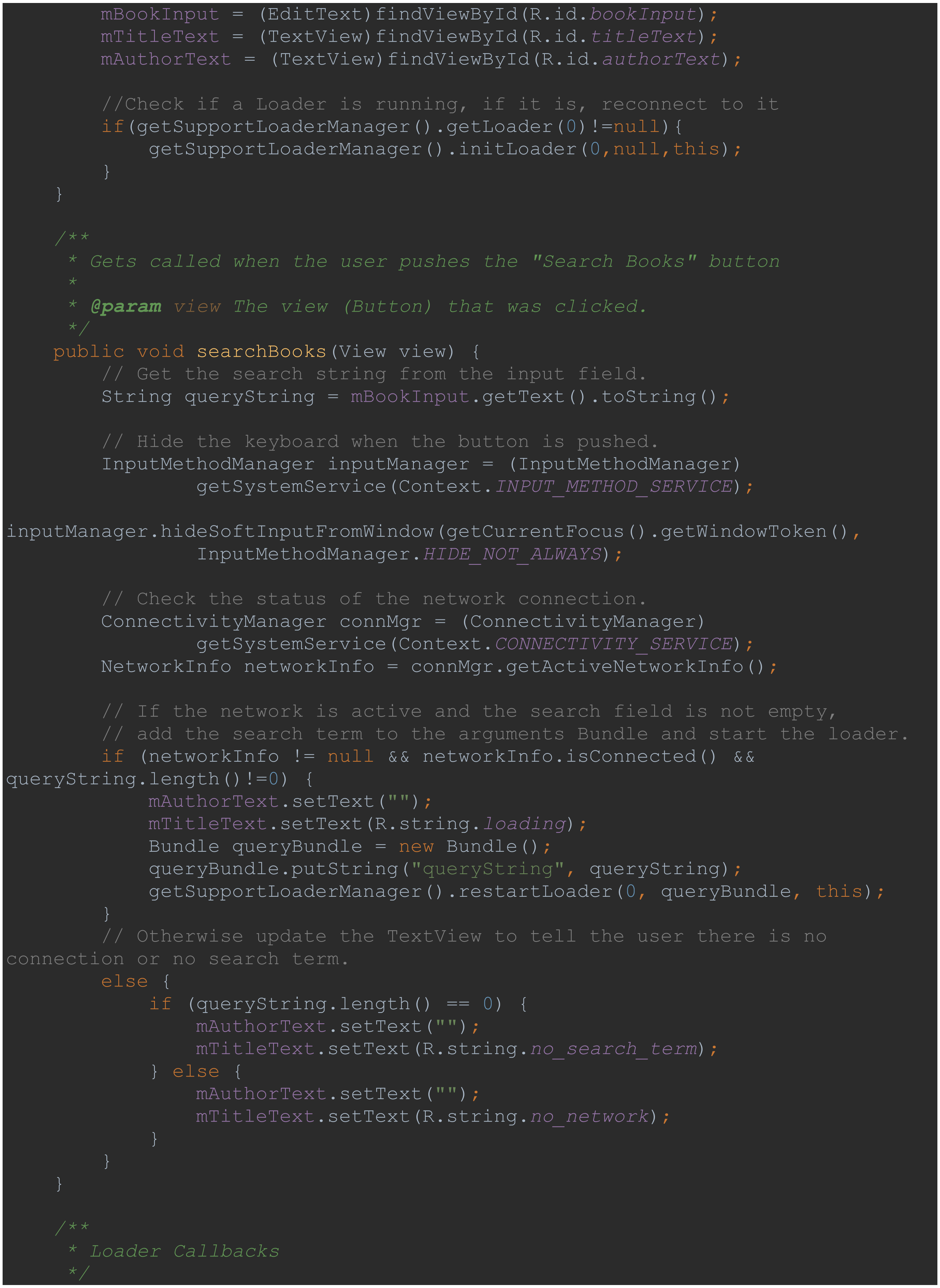
Woman's Doctor
["William J. Sweeney","Barbara Lang Stern"]

- **BookLeader.java**

```
package com.example.whowrotelt;

import android.content.Context;

import androidx.loader.content.AsyncTaskLoader;

public class BookLoader extends AsyncTaskLoader<String> {

    // Variable that stores the search string.
    private String mQueryString;

    // Constructor providing a reference to the search term.
    public BookLoader(Context context, String queryString) {
        super(context);
        mQueryString = queryString;
    }

    /**
     * This method is invoked by the LoaderManager whenever the loader is
started
     */
    @Override
    protected void onStartLoading() {
        forceLoad(); // Starts the loadInBackground method
    }

    /**
     * Connects to the network and makes the Books API request on a
background thread.
     *
     * @return Returns the raw JSON response from the API call.
     */
    @Override
    public String loadInBackground() {
        return NetworkUtils.getBookInfo(mQueryString);
    }
}
```

- **MainActivity.java**

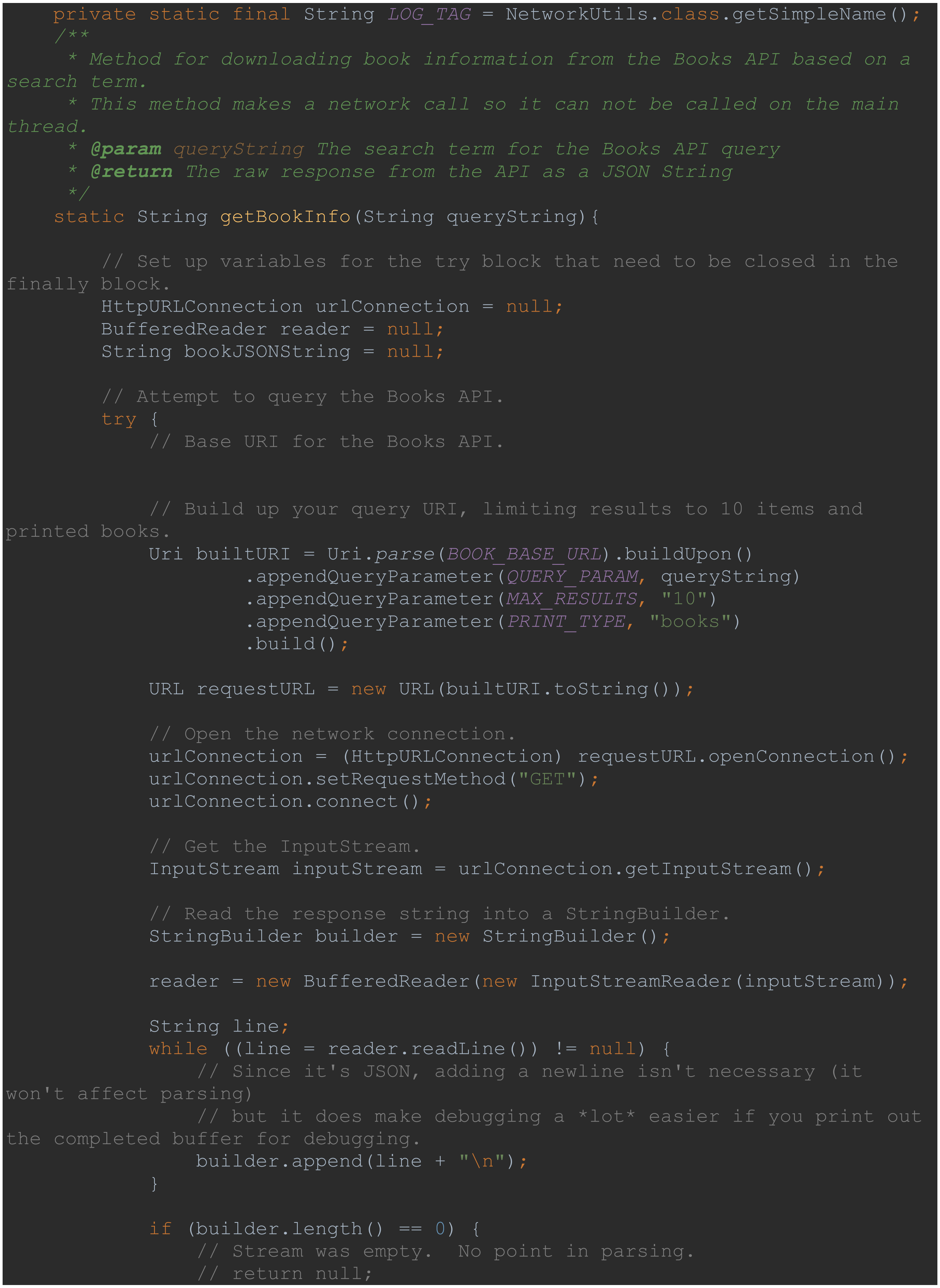
```
package com.example.whowroteIt;

import androidx.appcompat.app.AppCompatActivity;
import androidx.loader.app.LoaderManager;
import androidx.loader.content.Loader;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity implements
LoaderManager.LoaderCallbacks<String>{

    // Variables for the search input field, and results TextViews
    private EditText mBookInput;
    private TextView mTitleText;
    private TextView mAuthorText;


    /**
     * Initializes the activity.
     *
     * @param savedInstanceState The current state data
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize all the view variables
```



```
        mBookInput = (EditText)findViewById(R.id.bookInput);
        mTitleText = (TextView)findViewById(R.id.titleText);
        mAuthorText = (TextView)findViewById(R.id.authorText);

        //Check if a Loader is running, if it is, reconnect to it
        if(getSupportLoaderManager().getLoader(0)!=null){
            getSupportLoaderManager().initLoader(0,null,this);
        }
    }

    /**
     * Gets called when the user pushes the "Search Books" button
     *
     * @param view The view (Button) that was clicked.
     */
    public void searchBooks(View view) {
        // Get the search string from the input field.
        String queryString = mBookInput.getText().toString();

        // Hide the keyboard when the button is pushed.
        InputMethodManager inputManager = (InputMethodManager)
                getSystemService(Context.INPUT_METHOD_SERVICE);

inputManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),
                InputMethodManager.HIDE_NOT_ALWAYS);

        // Check the status of the network connection.
        ConnectivityManager connMgr = (ConnectivityManager)
                getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();

        // If the network is active and the search field is not empty,
        // add the search term to the arguments Bundle and start the loader.
        if (networkInfo != null && networkInfo.isConnected() &&
queryString.length()!=0) {
            mAuthorText.setText("");
            mTitleText.setText(R.string.loading);
            Bundle queryBundle = new Bundle();
            queryBundle.putString("queryString", queryString);
            getSupportLoaderManager().restartLoader(0, queryBundle, this);
        }
        // Otherwise update the TextView to tell the user there is no
connection or no search term.
        else {
            if (queryString.length() == 0) {
                mAuthorText.setText("");
                mTitleText.setText(R.string.no_search_term);
            } else {
                mAuthorText.setText("");
                mTitleText.setText(R.string.no_network);
            }
        }
    }

    /**
     * Loader Callbacks
     */
    /**
     * The LoaderManager calls this method when the loader is created.
     *
     * @param id ID integer to id   entify the instance of the loader.
     * @param args The bundle that contains the search parameter.
     * @return Returns a new BookLoader containing the search term.
     */
    @Override
    public Loader<String> onCreateLoader(int id, Bundle args) {
        return new BookLoader(this, args.getString("queryString"));
    }

    /**
     * Called when the data has been loaded. Gets the desired information
from
     * the JSON and updates the Views.
     *
     * @param loader The loader that has finished.
     * @param data The JSON response from the Books API.
     */
    @Override
    public void onLoadFinished(Loader<String> loader, String data) {
        try {
            // Convert the response into a JSON object.
            JSONObject jsonObject = new JSONObject(data);
            // Get the JSONArray of book items.
            JSONArray itemsArray = jsonObject.getJSONArray("items");

            // Initialize iterator and results fields.
            int i = 0;
            String title = null;
            String authors = null;

            // Look for results in the items array, exiting when both the
title and author
            // are found or when all items have been checked.
            while (i < itemsArray.length() || (authors == null && title ==
null)) {
                // Get the current item information.
                JSONObject book = itemsArray.getJSONObject(i);
                JSONObject volumeInfo = book.getJSONObject("volumeInfo");

                // Try to get the author and title from the current item,
                // catch if either field is empty and move on.
                try {
                    title = volumeInfo.getString("title");
                    authors = volumeInfo.getString("authors");
                } catch (Exception e){
                    e.printStackTrace();
                }

                // Move to the next item.
                i++;
            }

            // If both are found, display the result.
            if (title != null && authors != null){
                mTitleText.setText(title);
                mAuthorText.setText(authors);
                mBookInput.setText("");
            } else {
                // If none are found, update the UI to show failed results.
                mTitleText.setText(R.string.no_results);
                mAuthorText.setText("");
            }

        } catch (Exception e){
            // If onPostExecute does not receive a proper JSON string, update the UI to show failed results.
            mTitleText.setText(R.string.no_results);
            mAuthorText.setText("");
            e.printStackTrace();
        }


    }

    /**
     * In this case there are no variables to clean up when the loader is reset.
     *
     * @param loader The loader that was reset.
     */
    @Override
    public void onLoaderReset(Loader<String> loader) {}
}
```

- **NetworkUtils.java**

```
package com.example.whowroteit;

import android.net.Uri;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

class NetworkUtils {

    private static final String BOOK_BASE_URL = "https://www.googleapis.com/books/v1/volumes?";
    private static final String QUERY_PARAM = "q"; // Parameter for the search string.
    private static final String MAX_RESULTS = "maxResults"; // Parameter that limits search results.
    private static final String PRINT_TYPE = "printType"; // Parameter to filter by print type.

    // Class name for Log tag.
```



```
    private static final String LOG_TAG = NetworkUtils.class.getSimpleName();
    /**
     * Method for downloading book information from the Books API based on a 
search term.
     * This method makes a network call so it can not be called on the main 
thread.
     * @param queryString The search term for the Books API query
     * @return The raw response from the API as a JSON String
     */
    static String getBookInfo(String queryString){

        // Set up variables for the try block that need to be closed in the 
finally block.
        HttpURLConnection urlConnection = null;
        BufferedReader reader = null;
        String bookJSONString = null;

        // Attempt to query the Books API.
        try {
            // Base URI for the Books API.


            // Build up your query URI, limiting results to 10 items and 
printed books.
            Uri builtURI = Uri.parse(BOOK_BASE_URL).buildUpon()
                    .appendQueryParameter(QUERY_PARAM, queryString)
                    .appendQueryParameter(MAX_RESULTS, "10")
                    .appendQueryParameter(PRINT_TYPE, "books")
                    .build();

            URL requestURL = new URL(builtURI.toString());

            // Open the network connection.
            urlConnection = (HttpURLConnection) requestURL.openConnection();
            urlConnection.setRequestMethod("GET");
            urlConnection.connect();

            // Get the InputStream.
            InputStream inputStream = urlConnection.getInputStream();

            // Read the response string into a StringBuilder.
            StringBuilder builder = new StringBuilder();

            reader = new BufferedReader(new InputStreamReader(inputStream));

            String line;
            while ((line = reader.readLine()) != null) {
                // Since it's JSON, adding a newline isn't necessary (it 
won't affect parsing)
                // but it does make debugging a *lot* easier if you print out 
the completed buffer for debugging.
                builder.append(line + "\n");
            }

            if (builder.length() == 0) {
                // Stream was empty.  No point in parsing.
                // return null;
                return null;
            }
            bookJSONString = builder.toString();

            // Catch errors.
        } catch (IOException e) {
            e.printStackTrace();

            // Close the connections.
        } finally {
            if (urlConnection != null) {
                urlConnection.disconnect();
            }
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }

            }
        }

        // Return the raw response.
        return bookJSONString;
    }
}
```

- **Colors.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

- **Dimens.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>
</resources>
```

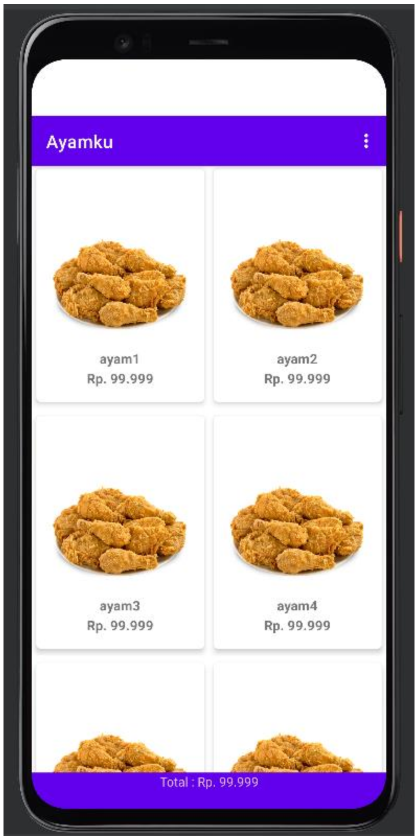
- **Strings.xml**

```xml
<resources>
    <string name="app_name">Who Wrote It?</string>
    <string name="instructions">Enter a book name, or part of a book name, or just some text from a book to find
    the full book title and who wrote the book!</string>
    <string name="button_text">Search Books</string>
    <string name="input_hint">Enter a Book Title</string>
    <string name="no_search_term">Please enter a search term</string>
    <string name="no_network">Please check your network connection and try again.</string>
    <string name="no_results">No Results Found</string>
    <string name="loading">Loading...</string>
</resources>
```

- **Styles.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

4. Aplikasi Penjualan Goreng “AyamKu”



- AndroidManifest.xml

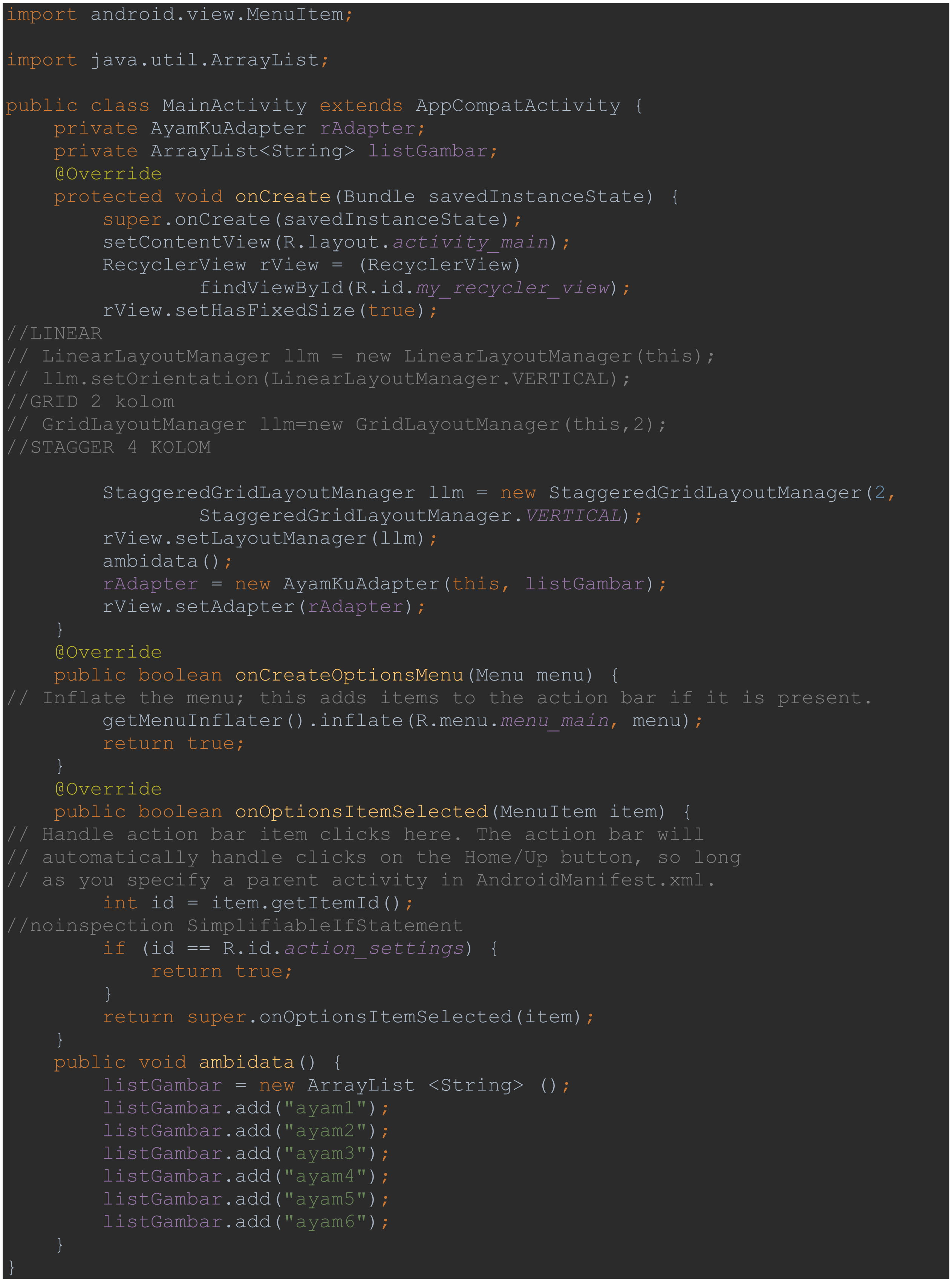
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.ryn.ayamku">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Ayamku"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- AyamkuAdapter.java

```
package com.example.ayamku;

import android.app.Activity;
import android.content.res.AssetManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

public class AyamKuAdapter extends
        RecyclerView.Adapter<AyamKuAdapter.ViewHolder> {
    public interface ItemClickListener {
        void onClick(View view, int position);
    }
    private ArrayList<String> listGambar;

    private Activity activity;
    private ItemClickListener clickListener;
    public void setClickLister(ItemClickListener clickListener) {
        this.clickListener = clickListener;
    }
    // Provide a reference to the views for each data item
// Complex data items may need more than one view per item, and
// you provide access to all the views for a data item in a view holder
    public class ViewHolder extends RecyclerView.ViewHolder implements
            View.OnClickListener {
        // each data item is just a string in this case
        private CardView cv;
        private TextView mTextView;
        private ImageView mImage;
        public ViewHolder(View v) {
            super(v);
            cv = (CardView) v.findViewById(R.id.card_view);
            mTextView = (TextView) v.findViewById(R.id.txt_card);
            mImage = (ImageView) v.findViewById(R.id.img_card);
            itemView.setTag(itemView);
            itemView.setOnClickListener(this);
        }
        public void onClick(View view) {
            if (clickListener != null) clickListener.onClick(view,
                    getAdapterPosition());
        }
    }
    // Provide a suitable constructor (depends on the kind of dataset)
    public AyamKuAdapter(Activity activity, ArrayList<String> listGambar) {
        this.listGambar = listGambar;
        this.activity = activity;
    }
    @Override
    public int getItemCount() {
        return listGambar.size();
    }
    // Create new views (invoked by the layout manager)
    @Override
    public AyamKuAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
// create a new view
        View v = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.item_list, parent, false);
// set the view's size, margins, paddings and layout parameters
        ViewHolder vh = new ViewHolder(v);
        return vh;
    }
    // Replace the contents of a view (invoked by the layout manager)
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
// - get element from your dataset at this position
// - replace the contents of the view with that element
        holder.mTextView.setText(listGambar.get(position));
// menampilkan gambar dari folder assets
        AssetManager assetManager = activity.getAssets();
        InputStream is;
        try {
            is = assetManager.open(listGambar.get(position) + ".png");
            Bitmap bitmap = BitmapFactory.decodeStream(is);

            System.out.print(listGambar.get(position) + ".png");
            holder.mImage.setImageBitmap(bitmap);
            is.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- MainActivity.java

```
package com.example.ayamku;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;

import android.os.Bundle;
import android.view.Menu;
```



```
import android.view.MenuItem;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    private AyamKuAdapter rAdapter;
    private ArrayList<String> listGambar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RecyclerView rView = (RecyclerView)
                findViewById(R.id.my_recycler_view);
        rView.setHasFixedSize(true);
//LINEAR
// LinearLayoutManager llm = new LinearLayoutManager(this);
// llm.setOrientation(LinearLayoutManager.VERTICAL);
//GRID 2 kolom
// GridLayoutManager llm=new GridLayoutManager(this,2);
//STAGGER 4 KOLOM

        StaggeredGridLayoutManager llm = new StaggeredGridLayoutManager(2,
                StaggeredGridLayoutManager.VERTICAL);
        rView.setLayoutManager(llm);
        ambidata();
        rAdapter = new AyamKuAdapter(this, listGambar);
        rView.setAdapter(rAdapter);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
// Handle action bar item clicks here. The action bar will
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
//noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
    public void ambidata() {
        listGambar = new ArrayList <String> ();
        listGambar.add("ayam1");
        listGambar.add("ayam2");
        listGambar.add("ayam3");
        listGambar.add("ayam4");
        listGambar.add("ayam5");
        listGambar.add("ayam6");
    }
}
```

- Activity_main.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"
    tools:context=".MainActivity">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/my_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical" />
    <TextView
        android:id="@+id/totalharga"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/my_recycler_view"
        android:layout_marginTop="-40dp"
        android:background="@color/purple_500"
        android:text="Total : Rp. 99.999"
        android:textAlignment="center"
        android:textColor="#E1BEE7" />
</RelativeLayout>
```

- Item_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:adroid="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <androidx.cardview.widget.CardView
        xmlns:card_view="http://schemas.android.com/apk/res-auto"
        android:id="@+id/card_view"
        android:layout_width="wrap_content"
        android:layout_height="273dp"
        android:layout_gravity="center"
        card_view:cardBackgroundColor="@color/cardview_light_background"
        card_view:cardCornerRadius="5dp"
        card_view:cardElevation="5dp"
        card_view:cardMaxElevation="1dp"
        card_view:cardPreventCornerOverlap="false"
        card_view:cardUseCompatPadding="true">
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="16dp">
            <ImageView
                android:id="@+id/img_card"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true"
                android:src="@drawable/ayam1" />
            <ImageView
                android:id="@+id/img_card1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true"
                 />
            <TextView
                android:id="@+id/txt_card"
                android:layout_width="wrap_content"
                android:layout_height="42dp"
                android:layout_alignParentBottom="true"
                android:layout_centerHorizontal="true"
                android:text="Ini Textview"
                android:textSize="15sp"
                android:textStyle="bold" />
            <TextView
                android:id="@+id/harga"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentBottom="true"
                android:layout_centerHorizontal="true"
                android:text="Rp. 99.999"
                android:textSize="15sp"
                android:textStyle="bold" />
        </RelativeLayout>
    </androidx.cardview.widget.CardView>
</LinearLayout>
```

- Menu_main.xml

```
<menu 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"
    tools:context="com.ryn.ayamku.MainActivity">
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        android:title="@string/action_settings"
        app:showAsAction="never" />
</menu>
```

- Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

- Strings.xml

```
<resources>
    <string name="app_name">Ayamku</string>
    <string name="action_settings">action</string>
</resources>
```