Rahul Nagaraju | **A20543969** | rnagaraju@hawk.iit.edu

Lab5 – Mar/30/2024

Intelligent Device Applications (ITMD-555)

PROJECT	Quiz App	**50 points**

Objective: This lab will have you create a quiz app via a question bank from the URL http://www.papademas.net:81/sample.txt. Check it.

Source Code:

MainActivity.java

```
package com.example.quiz;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.os.Bundle;

import android.os.AsyncTask;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RatingBar;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
```

```
import java.util.Collections;

public class MainActivity extends AppCompatActivity {

    TextView txtView;
    ArrayList<String> stringList = new ArrayList<String>();

    int correctAnswers = 0; // Track the number of correct answers
    long startTime; // To track the start time of the quiz
    static int questionNum = 0;

    private RadioGroup radioQuestions;
    private RadioButton radioButton;

    ImageView image;
    RatingBar ratingBar;
    private TextView timeTakenTextView;

    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        BackgroundTask bt = new BackgroundTask();
        bt.execute("http://www.papademas.net:81/sample.txt"); //grab url

        // Initialize the ProgressDialog
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Loading questions...");
        progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
        progressDialog.setMax(100);

        // Start the thread to fetch questions
        fetchQuestions();

        startTime = System.currentTimeMillis(); // Start the timer
```

```
}//end onCreate


// Method to fetch questions from the server
private void fetchQuestions() {
    // Show ProgressDialog
    progressDialog.show();

    // Start a new thread to fetch questions
    new Thread(new Runnable() {
        @Override
        public void run() {
            // Simulate fetching questions by sleeping for some time
            int progress = 0;
            while (progress < 100) {
                try {
                    Thread.sleep(50); // Simulate delay in fetching data
                    progress++;
                    progressDialog.setProgress(progress);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            // After fetching questions, dismiss the ProgressDialog and start the quiz
            progressDialog.dismiss();
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    // Shuffle the questions
                    Collections.shuffle(stringList);
                    // Start the quiz
                    //startQuiz();
                    displayFirstQuestion();
                }
            });
        }
    }).start();
}
```

```
//background process to download the file from internet
private class BackgroundTask extends AsyncTask<String, Integer, Void> {

    protected void onPreExecute() {  }

    protected Void doInBackground(String... params) {
        URL url;
        String StringBuffer = null;
        try {
            //create url object to point to the file location on internet
            url = new URL(params[0]);
            //make a request to server
            HttpURLConnection con = (HttpURLConnection) url.openConnection();
            //get InputStream instance
            InputStream is = con.getInputStream();
            //create BufferedReader object
            BufferedReader br = new BufferedReader(new InputStreamReader(is));

            //read content of the file line by line & add it to Stringbuffer
            while ((StringBuffer = br.readLine()) != null) {
                stringList.add(StringBuffer);  //add to Arraylist
            }

            br.close();

        } catch (Exception e) { e.printStackTrace(); }
        return null;
    }

    protected void onPostExecute(Void result) {
        txtView = findViewById(R.id.textView1);
        //display read text in TextVeiw
        txtView.setText(stringList.get(0));
        startQuiz();
    }
}//end BackgroundTask class

// Method to display the first question after shuffling
private void displayFirstQuestion() {
```

```
        // Display the first question if the list is not empty
        if (!stringList.isEmpty()) {
            txtView.setText(stringList.get(0));
            startQuiz();
        } else {
            txtView.setText("No questions for today!!!");
        }
    }

    public void startQuiz() {
        buttonListener();
    }

    public void buttonListener() {

        Button btnDisplay;

        radioQuestions = findViewById(R.id.radioQuestions);
        btnDisplay = findViewById(R.id.btnDisplay);

        btnDisplay.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                // get selected radio button from radioGroup
                int selectedId = radioQuestions.getCheckedRadioButtonId();

                // find the radiobutton by returned id
                radioButton = findViewById(selectedId);

                // Increment correctAnswers if the selected answer is correct
                if ((questionNum == 0 || questionNum == 2 || questionNum == 3) &&
radioButton.getText().equals("True") ||
                        (questionNum == 1 || questionNum == 4) && radioButton.getText().equals("False")) {
                    correctAnswers++;
                }

                switch (questionNum) {
```

```
        case 0:
        case 2:
        case 3:

            //verify if result matches the right button selection
            //i.e., (True or false!)
            if (radioButton.getText().equals("True"))
                Toast.makeText(MainActivity.this,
                        " Right!", Toast.LENGTH_SHORT).show();
            else
                Toast.makeText(MainActivity.this,
                        " Wrong!", Toast.LENGTH_SHORT).show();
            break;

        case 1:
        case 4:

            //verify if result matches the right button selection
            //i.e., (True or false!)
            if (radioButton.getText().equals("False"))
                Toast.makeText(MainActivity.this,
                        " Right!", Toast.LENGTH_SHORT).show();
            else
                Toast.makeText(MainActivity.this,
                        " Wrong!", Toast.LENGTH_SHORT).show();
            break;

        //finish switch cases 2-4
    }//end switch


    // If it's the last question, show RatingBar and calculate elapsed time
    if (questionNum == stringList.size() - 1) {
        long endTime = System.currentTimeMillis();
        long elapsedTime = endTime - startTime;

        displayElapsedTime(elapsedTime);
```

```
                ratingBar=findViewById(R.id.ratingBar);
                displayRating();
                ratingBar.setVisibility(View.VISIBLE);
            }
        }
    });
    imageListener();
}//end buttonListener

// Helper method to display elapsed time
private void displayElapsedTime(long elapsedTime) {
    // Convert milliseconds to seconds
    long elapsedSeconds = elapsedTime / 1000;
    // Display elapsed time in some TextView or LogCat
    // Initialize the TextView for displaying time taken

    timeTakenTextView = findViewById(R.id.timeTakenTextView);
    timeTakenTextView.setVisibility(View.VISIBLE);
    timeTakenTextView.setText("Time taken: " + elapsedSeconds + " seconds");
    Log.d("Quiz", "Elapsed Time: " + elapsedSeconds + " seconds");
}

// Helper method to display the rating based on correct answers
private void displayRating() {
    // Calculate the percentage of correct answers
    int totalQuestions = stringList.size();
    float percentageCorrect = (float) correctAnswers / totalQuestions * 100;

    // Set the rating based on the percentage of correct answers
    float rating = percentageCorrect / 20; // Since RatingBar ranges from 0 to 5
    ratingBar.setRating(rating);
}
public void imageListener() {

    image = findViewById(R.id.imageView1);
    image.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
```

```
            // get new question for viewing
            if (questionNum == 4)
                //reset count to -1 to start first question again
                questionNum = -1;
            txtView.setText(stringList.get(++questionNum));
            //reset radio button (radioTrue) to default
            radioQuestions.check(R.id.radioTrue);
        }
    });
  }//end imageListener
}//end activity
```

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:orientation="vertical" >

  <TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:width="400dp"
    android:paddingTop="20px"
    android:paddingBottom="20px"
    android:textSize="12sp" />

  <RadioGroup
    android:id="@+id/radioQuestions"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
```

```
        android:id="@+id/radioTrue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_true"
        android:checked="true" />

    <RadioButton
        android:id="@+id/radioFalse"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/radio_false" />

</RadioGroup>

<Button
    android:id="@+id/btnDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/btn_display" />

<ImageView
    android:id="@+id/imageView1"
    android:layout_width="82dp"
    android:layout_height="66dp"
    android:clickable="true"
    android:src="@drawable/next" />

<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:stepSize="1.0"
    android:rating="2.0"
    android:isIndicator="true"
    android:visibility="invisible"/>

<TextView
    android:id="@+id/timeTakenTextView"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:padding="8dp"
        android:textColor="@android:color/black"
        android:visibility="gone" />

</LinearLayout>
```

Android Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:usesCleartextTraffic="true"
        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.Quiz"
        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>
```

strings.xml

```
<resources>
    <string name="app_name">Quiz</string>
    <string name="radio_true">True</string>
    <string name="radio_false">False</string>
    <string name="btn_display">Display Results</string>
</resources>
```