**Міністерствоосвітиі наукиУкраїни**

**Національнийуніверситет «Львівська політехніка»**

**Інститут телекомунікацій, радіоелектроніки та електронноїтехніки**



Звіт з лабораторної роботи

з дисципліни «Розробка мобільних додатків»

На тему: «Розробка мобільного додатку»

Виконав:

ст. гр. ТР-45

Пилип С.С.

Прийняв:

Андрущак В.С.

Львів 2022

**Хід роботи:**

**Розробка музичного плеєра для OC Android в середовищі Android Studio на Java.**

1. Візуальна складова, з якою буде взаємодіяти користувач, описана в двох документах формату xml.

В документі activity\_main.xml описана візуальна структура програми (порядок розміщення елементів керування, розмір елементів, колір елементів та фон всього додатку).

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">  
  
  
 <LinearLayout  
 android:id="@+id/bottom"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/bacground"  
 android:orientation="vertical"  
 android:padding="1dp"  
 >  
  
 <ImageView  
 android:id="@+id/audio\_art"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_marginTop="20dp"  
 android:src="@drawable/image" />  
  
 <TextView  
 android:id="@+id/audio\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:ellipsize="end"  
 android:textAlignment="center"  
 android:textColor="@color/colorAccent"  
 android:textSize="16sp" />  
 <TextView  
 android:id="@+id/audio\_artist"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:ellipsize="end"  
 android:textAlignment="center"  
 android:textColor="@color/colorAccent"  
 android:textSize="14sp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginLeft="2dp"  
 android:layout\_marginRight="2dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/current"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginEnd="2dp"  
 android:text="00:00"  
 android:textColor="@color/colorAccent" />  
  
 <SeekBar  
 android:id="@+id/seekbar"  
 style="@style/Base.Widget.AppCompat.SeekBar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:theme="@style/SeekBar" />  
  
 <TextView  
 android:id="@+id/total"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginStart="2dp"  
 android:layout\_marginEnd="2dp"  
 android:text="00:00"  
 android:textColor="@color/colorAccent" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="10dp"  
 android:gravity="center"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/prev"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:layout\_marginRight="20dp"  
 android:src="@drawable/ic\_skip\_previous\_black\_24dp" />  
  
 <ImageView  
 android:id="@+id/pause"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:src="@drawable/ic\_play\_circle\_filled\_black\_24dp" />  
  
 <ImageView  
 android:id="@+id/next"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:layout\_marginLeft="20dp"  
 android:src="@drawable/ic\_skip\_next\_black\_24dp" />  
  
 </LinearLayout>

<android.support.v7.widget.RecyclerView  
 android:id="@+id/recycler\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/bacground"  
 />  
 </LinearLayout>  
  
</RelativeLayout>

В документі audio\_list.xml описана структура для відображення інформації для кожного аудіо файлу (назва треку, виконавець і обкладинка альбому). Перелік аудіофайлів має вигляд списку

audio\_list.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="@drawable/bacground">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:layout\_marginTop="1dp"  
 android:background="@color/background">  
  
 <ImageView  
 android:id="@+id/image"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_centerVertical="true"  
 android:layout\_marginStart="10dp"  
 android:background="@drawable/background\_gradient"  
 android:src="@drawable/ic\_audiotrack\_black\_24dp" />  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="55dp"  
 android:layout\_centerVertical="true"  
 android:layout\_marginStart="60dp">  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="25sp"  
 android:layout\_marginTop="2dp"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:textColor="@color/colorAccent"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/artist"  
 android:layout\_width="match\_parent"  
 android:layout\_height="25sp"  
 android:layout\_marginTop="30dp"  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:textColor="@color/colorAccent"  
 android:textSize="14sp" />  
 </RelativeLayout>  
 </RelativeLayout>  
</RelativeLayout>

2. Вся логіка додатку написана в трьох файлах формату .java.

У файлі MainActivity.java є головним файлом у якому створюються змінні для керування кнопками та усім інтерфейсом додатку. Без цього файлу додаток не мав би функціоналу і виглядав як статичне зображення.

MainActivity.java

import android.Manifest;  
import android.content.ContentResolver;  
import android.content.ContentUris;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.media.MediaPlayer;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Handler;  
import android.provider.MediaStore;  
import android.support.v4.app.ActivityCompat;  
import android.support.v4.content.ContextCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.DefaultItemAnimator;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.SeekBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
  
 ArrayList<ModelAudio> audioArrayList;  
 RecyclerView recyclerView;  
 MediaPlayer mediaPlayer;  
 double current\_pos, total\_duration;  
 TextView current, total, audio\_name, audio\_artist;  
 ImageView audio\_art;  
 ImageView prev, next, play;  
 SeekBar seekBar;  
 int audio\_index = 0;  
 public static final int *PERMISSION\_READ* = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 if (checkPermission()) {  
 setAudio();  
 }  
 }  
  
 public void setAudio() {  
 recyclerView = (RecyclerView) findViewById(R.id.*recycler\_view*);  
 recyclerView.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.*VERTICAL*, false));  
 recyclerView.setItemAnimator(new DefaultItemAnimator());  
  
 current = (TextView) findViewById(R.id.*current*);  
 total = (TextView) findViewById(R.id.*total*);  
 audio\_name = (TextView) findViewById(R.id.*audio\_name*);  
 audio\_artist = (TextView) findViewById(R.id.*audio\_artist*);  
 audio\_art = (ImageView) findViewById(R.id.*audio\_art*);  
 prev = (ImageView) findViewById(R.id.*prev*);  
 next = (ImageView) findViewById(R.id.*next*);  
 play = (ImageView) findViewById(R.id.*pause*);  
 seekBar = (SeekBar) findViewById(R.id.*seekbar*);  
  
 audioArrayList = new ArrayList<>();  
 mediaPlayer = new MediaPlayer();  
  
 getAudioFiles();  
  
 //seekBar change listener  
 seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
 current\_pos = seekBar.getProgress();  
 mediaPlayer.seekTo((int) current\_pos);  
 }  
 });  
  
 mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {  
 @Override  
 public void onCompletion(MediaPlayer mp) {  
 audio\_index++;  
 if (audio\_index < (audioArrayList.size())) {  
 playAudio(audio\_index);  
 } else {  
 audio\_index = 0;  
 playAudio(audio\_index);  
 }  
  
 }  
 });  
  
 if (!audioArrayList.isEmpty()) {  
 playAudio(audio\_index);  
 prevAudio();  
 nextAudio();  
 setPause();  
 }  
 }  
  
 //play audio file  
 public void playAudio(int pos) {  
 try {  
 mediaPlayer.reset();  
 //set file path  
 mediaPlayer.setDataSource(this, audioArrayList.get(pos).getAudioUri());  
 mediaPlayer.prepare();  
 //mediaPlayer.start();  
 play.setImageResource(R.drawable.*ic\_play\_circle\_filled\_black\_24dp*);  
 audio\_name.setText(audioArrayList.get(pos).getAudioTitle());  
 audio\_artist.setText(audioArrayList.get(pos).getAudioArtist());  
 audio\_art.setImageURI(audioArrayList.get(pos).audioArtUri);  
 audio\_index = pos;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 setAudioProgress();  
 }  
  
 //set audio progress  
 public void setAudioProgress() {  
 //get the audio duration  
 current\_pos = mediaPlayer.getCurrentPosition();  
 total\_duration = mediaPlayer.getDuration();  
  
 //display the audio duration  
 total.setText(timerConversion((long) total\_duration));  
 current.setText(timerConversion((long) current\_pos));  
 seekBar.setMax((int) total\_duration);  
 final Handler handler = new Handler();  
  
 Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 try {  
 current\_pos = mediaPlayer.getCurrentPosition();  
 current.setText(timerConversion((long) current\_pos));  
 seekBar.setProgress((int) current\_pos);  
 handler.postDelayed(this, 1000);  
 } catch (IllegalStateException ed) {  
 ed.printStackTrace();  
 }  
 }  
 };  
 handler.postDelayed(runnable, 1000);  
 }  
  
 //play previous audio  
 public void prevAudio() {  
 prev.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (audio\_index > 0) {  
 audio\_index--;  
 playAudio(audio\_index);  
 } else {  
 audio\_index = audioArrayList.size() - 1;  
 playAudio(audio\_index);  
 }  
 }  
 });  
 }  
  
 //play next audio  
 public void nextAudio() {  
 next.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (audio\_index < (audioArrayList.size() - 1)) {  
 audio\_index++;  
 playAudio(audio\_index);  
 } else {  
 audio\_index = 0;  
 playAudio(audio\_index);  
 }  
 }  
 });  
 }  
  
 //pause audio  
 public void setPause() {  
 play.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mediaPlayer.isPlaying()) {  
 mediaPlayer.pause();  
 play.setImageResource(R.drawable.*ic\_play\_circle\_filled\_black\_24dp*);  
 } else {  
 mediaPlayer.start();  
 play.setImageResource(R.drawable.*ic\_pause\_circle\_filled\_black\_24dp*);  
 }  
 }  
 });  
 }  
  
 //time conversion  
 public String timerConversion(long value) {  
 String audioTime;  
 int dur = (int) value;  
 int hrs = (dur / 3600000);  
 int mns = (dur / 60000) % 60000;  
 int scs = dur % 60000 / 1000;  
  
 if (hrs > 0) {  
 audioTime = String.*format*("%02d:%02d:%02d", hrs, mns, scs);  
 } else {  
 audioTime = String.*format*("%02d:%02d", mns, scs);  
 }  
 return audioTime;  
 }  
  
 //fetch the audio files from storage  
 public void getAudioFiles() {  
 ContentResolver contentResolver = getContentResolver();  
 Uri uri = MediaStore.Audio.Media.*EXTERNAL\_CONTENT\_URI*;  
  
 Cursor cursor = contentResolver.query(uri, null, null, null, null);  
  
 //looping through all rows and adding to list  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 String title = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*TITLE*));  
 String artist = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*ARTIST*));  
 String duration = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DURATION*));  
 String url = cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.*DATA*));  
 int albumId = (int) cursor.getLong(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.*ALBUM\_ID*));  
 Uri sArtworkUri = Uri.*parse*("content://media/external/audio/albumart");  
 Uri albumArtUri = ContentUris.*withAppendedId*(sArtworkUri, albumId);  
  
 ModelAudio modelAudio = new ModelAudio();  
  
 modelAudio.setAudioTitle(title);  
 modelAudio.setAudioArtist(artist);  
 modelAudio.setAudioUri(Uri.*parse*(url));  
 modelAudio.setAudioDuration(duration);  
 modelAudio.setAudioArtUri(albumArtUri);  
  
 audioArrayList.add(modelAudio);  
  
 } while (cursor.moveToNext());  
 }  
  
 AudioAdapter adapter = new AudioAdapter(this, audioArrayList);  
 recyclerView.setAdapter(adapter);  
  
 adapter.setOnItemClickListener(new AudioAdapter.OnItemClickListener() {  
 @Override  
 public void onItemClick(int pos, View v) {  
 playAudio(pos);  
 }  
 });  
 }  
  
 //runtime storage permission  
 public boolean checkPermission() {  
 int READ\_EXTERNAL\_PERMISSION = ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_EXTERNAL\_STORAGE*);  
 if ((READ\_EXTERNAL\_PERMISSION != PackageManager.*PERMISSION\_GRANTED*)) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*}, *PERMISSION\_READ*);  
 return false;  
 }  
 return true;  
 }  
  
 public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 switch (requestCode) {  
 case *PERMISSION\_READ*: {  
 if (grantResults.length > 0 && permissions[0].equals(Manifest.permission.*READ\_EXTERNAL\_STORAGE*)) {  
 if (grantResults[0] == PackageManager.*PERMISSION\_DENIED*) {  
 Toast.*makeText*(getApplicationContext(), "Please allow storage permission", Toast.*LENGTH\_LONG*).show();  
 } else {  
 setAudio();  
 }  
 }  
 }  
 }  
 }  
  
 //release mediaPlayer  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 if (mediaPlayer != null) {  
 mediaPlayer.release();  
 }  
 }  
}

У файлі AudioAdapter.java написана логіка для коректної роботи списку аудіофайлів. для кожного файлу відображається його назва та виконавець, і починається відтворення при натисканні на елемент списку який відповідає певному файлу.

AudioAdapter.java

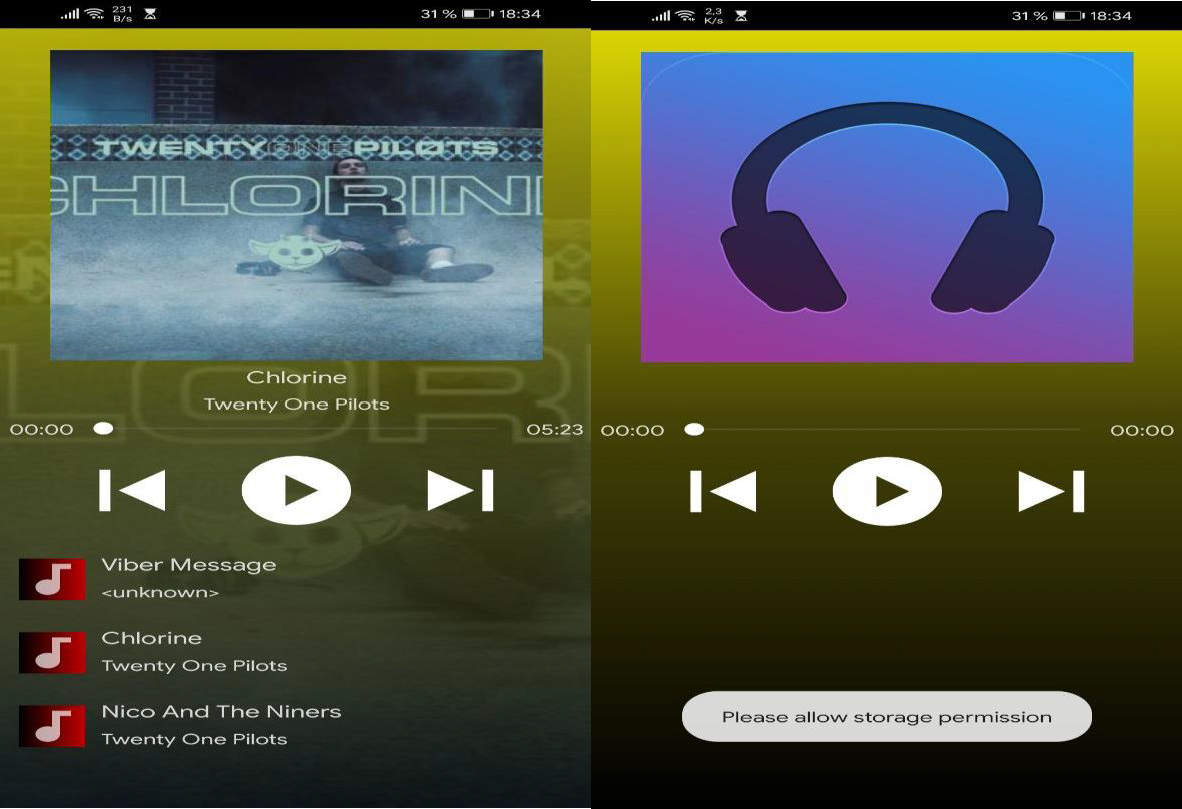
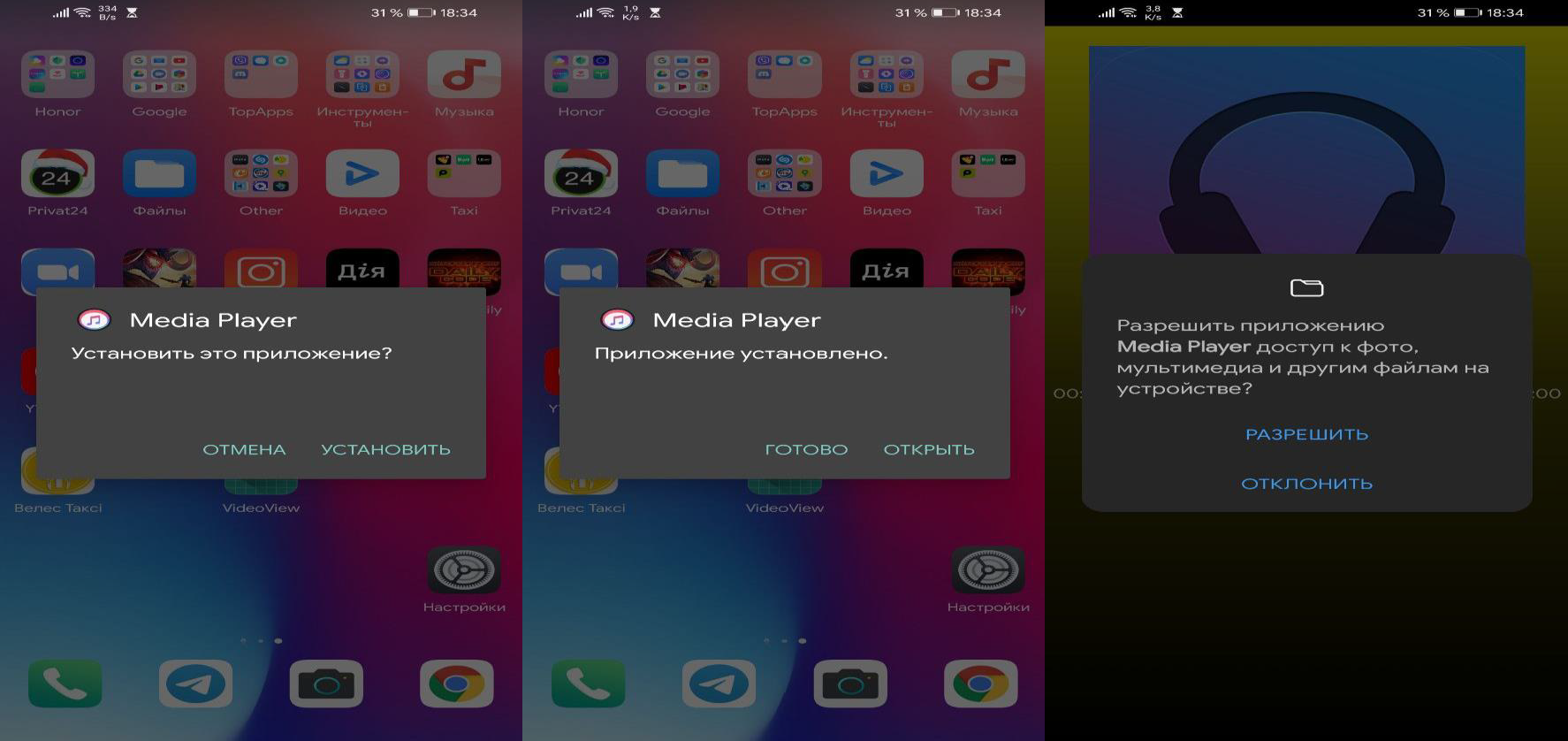
import android.content.Context;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
import java.util.ArrayList;  
  
public class AudioAdapter extends RecyclerView.Adapter<AudioAdapter.viewHolder> {  
  
 Context context;  
 ArrayList<ModelAudio> audioArrayList;  
 public OnItemClickListener onItemClickListener;  
  
 public AudioAdapter(Context context, ArrayList<ModelAudio> audioArrayList) {  
 this.context = context;  
 this.audioArrayList = audioArrayList;  
 }  
  
 @Override  
 public AudioAdapter.viewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {  
 View view = LayoutInflater.*from*(context).inflate(R.layout.*audio\_list*, viewGroup, false);  
 return new viewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(final AudioAdapter.viewHolder holder, final int i) {  
 holder.title.setText(audioArrayList.get(i).getAudioTitle());  
 holder.artist.setText(audioArrayList.get(i).getAudioArtist());  
 }  
  
 @Override  
 public int getItemCount() {  
 return audioArrayList.size();  
 }  
  
 public class viewHolder extends RecyclerView.ViewHolder {  
 TextView title, artist;  
 ImageView delete, edit;  
 public viewHolder(View itemView) {  
 super(itemView);  
 title = (TextView) itemView.findViewById(R.id.*title*);  
 artist = (TextView) itemView.findViewById(R.id.*artist*);  
  
 itemView.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 onItemClickListener.onItemClick(getAdapterPosition(), v);  
 }  
 });  
 }  
 }  
  
 public void setOnItemClickListener(OnItemClickListener onItemClickListener) {  
 this.onItemClickListener = onItemClickListener;  
 }  
  
 public interface OnItemClickListener {  
 void onItemClick(int pos, View v);  
 }  
}

Файл ModelAudio.java створений для вичитування і зберігання інформації про аудіофайл. Коли програма знаходить файл формату .mp3 на смартфоні, то створюється об’єкт класу ModelAudio в який записується вся інформація про файл. Після створення об’єкту, який тримає в собі інформацію про файл, він додається до колекціїї ArrayList. Об’єкт класу ArrayList створюється в класі MainActivity і містить в собі масив об’єктів ModelAudio.

ModelAudio.java

import android.net.Uri;  
  
public class ModelAudio {  
  
 String audioTitle;  
 String audioDuration;  
 String audioArtist;  
 Uri audioUri;  
 Uri audioArtUri;  
  
 public String getAudioTitle() {  
 return audioTitle;  
 }  
  
 public void setAudioTitle(String audioTitle) {  
 this.audioTitle = audioTitle;  
 }  
  
 public String getAudioDuration() {  
 return audioDuration;  
 }  
  
 public void setAudioDuration(String audioDuration) {  
 this.audioDuration = audioDuration;  
 }  
  
 public String getAudioArtist() {  
 return audioArtist;  
 }  
  
 public void setAudioArtist(String audioArtist) {  
 this.audioArtist = audioArtist;  
 }  
  
 public Uri getAudioUri() {  
 return audioUri;  
 }  
  
 public void setAudioUri(Uri audioUri) {  
 this.audioUri = audioUri;  
 }  
  
 public void setAudioArtUri(Uri audioArtUri) { this.audioArtUri = audioArtUri; }  
  
 public Uri getAudioArtUri() { return audioArtUri; }  
  
}

## Для встановлення додатку просто потрібно відкрити .apk-файл на смартфоні та дати дозвіл на встановлення.



## Після встановлення додаток попросить дозвіл на доступ до медіа файлів на смартфоні.

## Якщо додаток отримає дозвіл, то в плеєрі будуть відображатись усі аудіо файли які існують на смартфоні. Якщо користувач не дозволить зчитування аудіо файлів, то додаток повідомить про заборону і виведе повідомлення з проханням дозволити доступ.