Vidzemes Augstskola

**Inženierzinātņu Fakultāte**

mobilo tehnoloģiju risinājumi

grupa 02

1 Mājas darbs

Valmiera, 2018

SATURS

[1 Ievads 4](#_Toc530477260)

[2 Kameras aktivitāte 4](#_Toc530477261)

[3 Audio ieraksta aktivitāte 5](#_Toc530477262)

[PIELIKUMS 1 6](#_Toc530477263)

[Pielikums 2 7](#_Toc530477264)

[Pielikums 3 10](#_Toc530477265)

[Pielikums 4 13](#_Toc530477266)

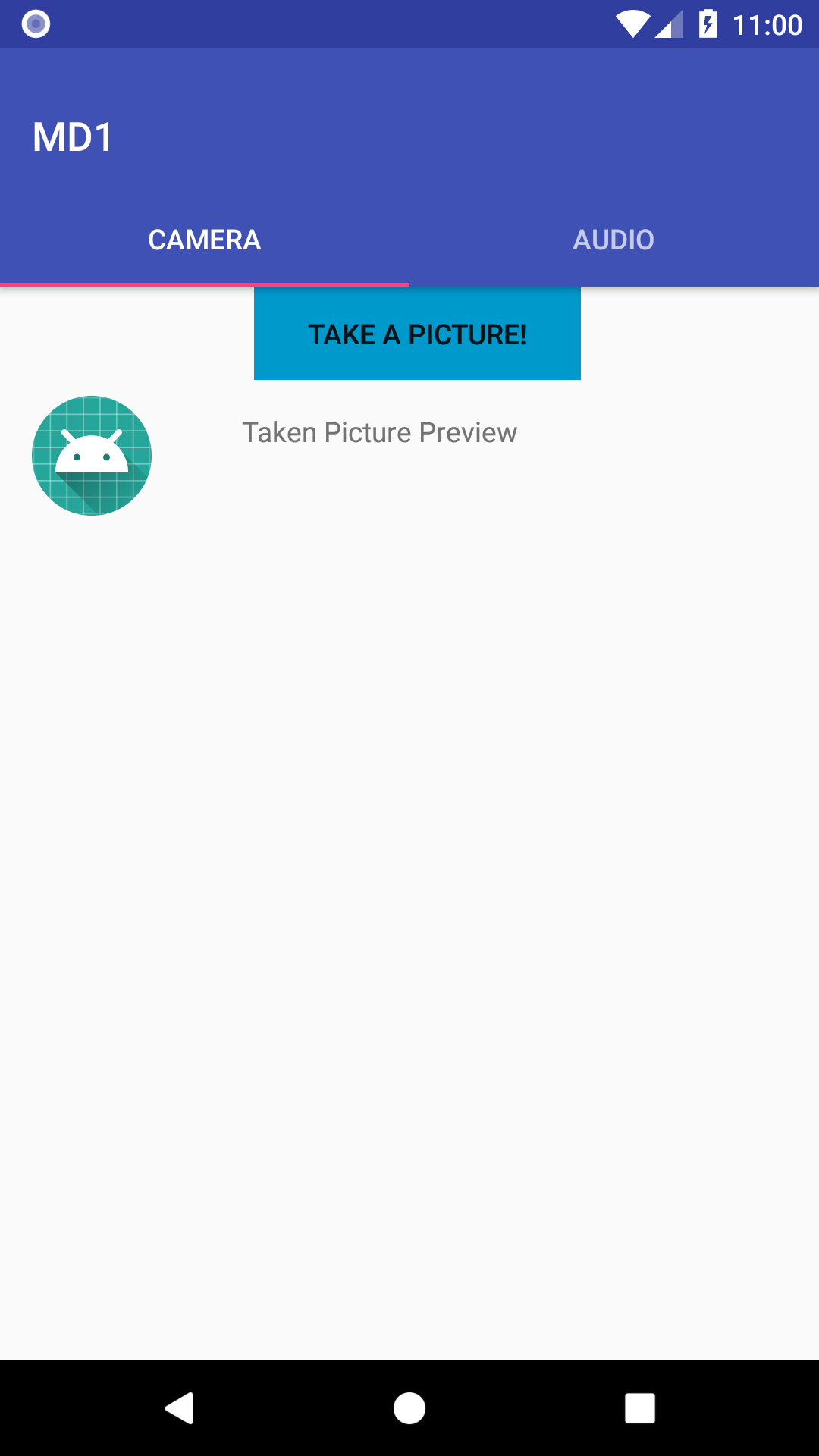
|  |  |  |  |
| --- | --- | --- | --- |
| Dokumenta vēsture | | | |
| Versija | Statuss / Izmaiņas | Datums | Autors |
| 1.0 | Pirmā versija | 19.01.2018 | Toms Amsons |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Kontakti un atbildīgais (-ie) | | | |
| Vārds, Uzvārds | Struktūrvienība | Amats | Kontaktinformācija (e-pasts) |
| Toms Amsons | Grupa #2 | Dalībnieks | toms.amsons@va.lv |
| Ģirts Ķesteris | Grupa #2 | Dalībnieks | girts.kesteris@va.lv |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Ievads

1. Mājas darbā tika izveidota lietotne, ar kuru var uzņemt fotogrāfijas un ierakstīt audio. Faili tiek saglabāti lokāli. Izslēdzot lietotni faili tiek dzēsti.

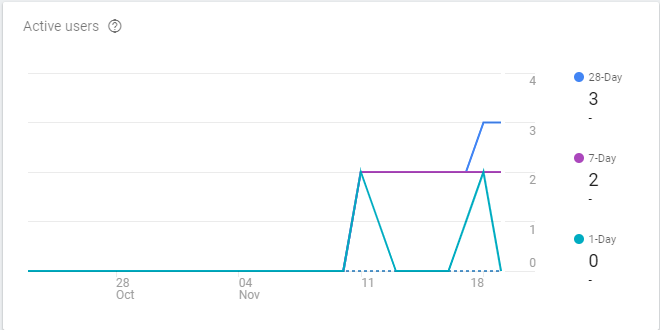
# Kameras aktivitāte



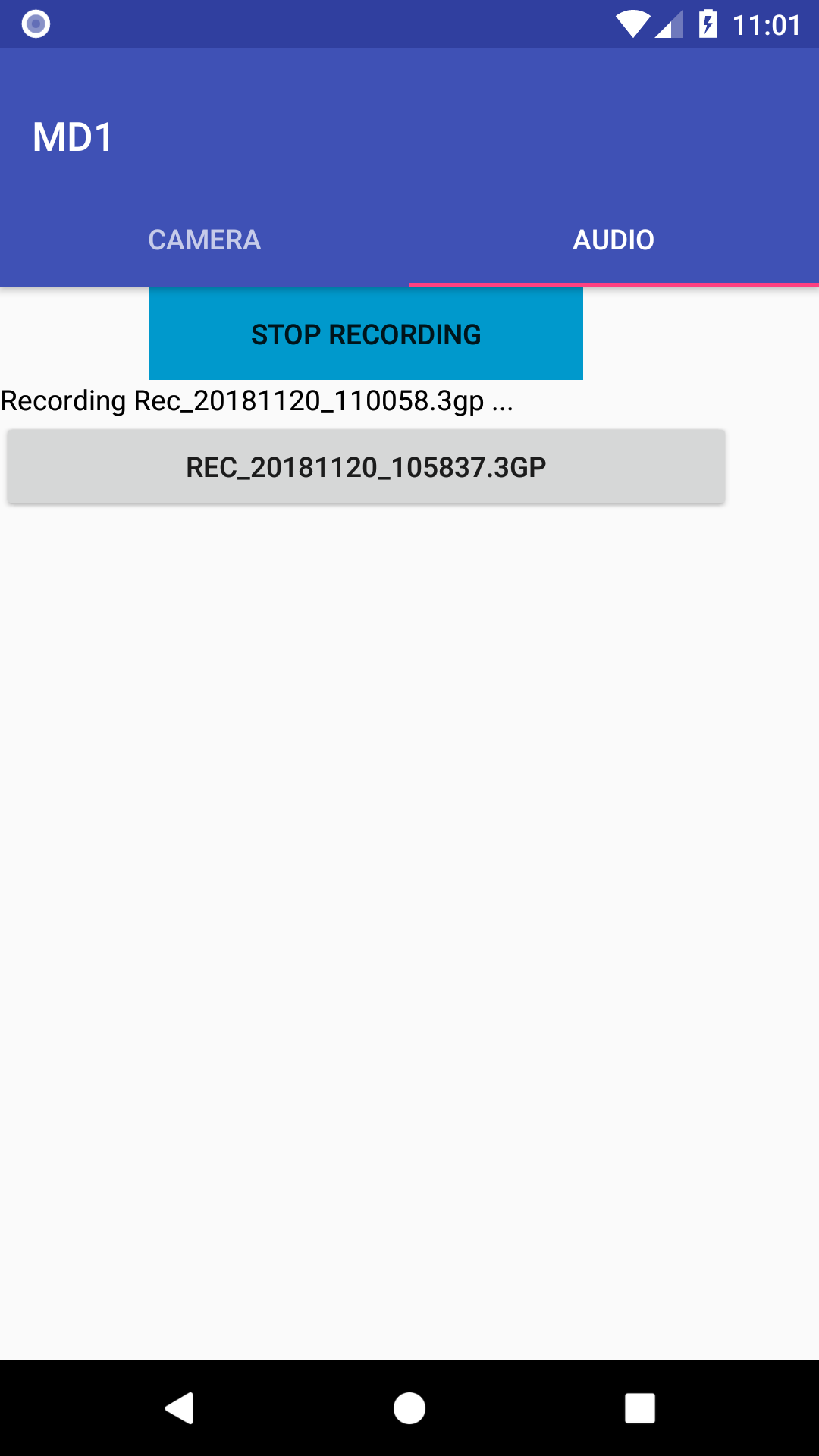
Atverot lietotni atveras pirmā aktivitāte, kas ir kameras aktivitāte. Šeit lietotājam ir iespēja uzņemt bildi, kā arī pārskatīt uzņemtās bildes. No šī skata lietotājam ir iespējams pārslēgties uz audio ieraksta skatu.

Šis skats tiek izveidots pēc “content\_fragment\_video.xml” (skat. 1 pielikumu) faila. Tiek definēta poga un režģa skats attēlu aplūkošanai.

Funkcionalitāte šim logam tiek definēta “MainActivity.java”(skat 2. pielikumu). Šajā failā tiek tiek izsaukts google firebase, lai iegūtu lietotnes apmeklējuma statistiku.



# Audio ieraksta aktivitāte



Šajā aktivitātē lietotājam ir iespēja ierakstīt audio failus, kā arī to atskaņot. Pēc ieraksta veikšanas, tas tiek attēlots zem ieraksta pogas saraksta veidā. Lai lietotājam būtu skaidrs vai ieraksts notiek, tiek attēlots teksts “Recording \*faila nosaukums\*”. Šie ieraksti tiek dzēsti, kad lietotne tiek aizvērta.

Šīs aktivitātes skats tiek veidots pēc “fragment\_audio.xml” faila. Tajā tiek definēts režģa skats audio ierakstu atskaņošanai, poga, lai sāktu un apstādinātu ierakstu.

# PIELIKUMS 1

Content\_fragment\_video.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context=".CameraFragment"  
 tools:showIn="@layout/activity\_main">  
  
 <Button  
 android:id="@+id/recordAudioButton"  
 style="@android:style/Widget.DeviceDefault.Button.Borderless"  
 android:layout\_width="164dp"  
 android:layout\_height="47dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="20dp"  
 android:background="@android:color/holo\_blue\_dark"  
 android:drawablePadding="8dp"  
 android:onClick="catchImage"  
 android:text="@string/catch\_image"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:padding="8dp"  
 tools:text="@string/catch\_image" />  
  
 <ImageView  
 android:id="@+id/thumbView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="16dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="8dp"  
 android:adjustViewBounds="true"  
 android:cropToPadding="true"  
 android:maxWidth="60dp"  
 android:scaleType="fitXY"  
 android:src="@mipmap/ic\_launcher"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/recordAudioButton"  
 tools:maxHeight="60dp" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="260dp"  
 android:layout\_height="44dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginLeft="24dp"  
 android:layout\_marginRight="8dp"  
 android:layout\_marginStart="24dp"  
 android:layout\_marginTop="16dp"  
 android:ems="10"  
 android:text="@string/image\_thumb"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/thumbView"  
 app:layout\_constraintTop\_toBottomOf="@+id/recordAudioButton" />  
  
  
 <HorizontalScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="250dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/thumbView">  
  
 <FrameLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:id="@+id/linearLayout\_gridtableLayout"  
 android:layout\_width="500dp"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
 <GridView  
 android:id="@+id/gridview"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="fill\_parent"  
 android:layout\_margin="4dp"  
 android:columnWidth="220dp"  
 android:gravity="center"  
 android:horizontalSpacing="1dp"  
 android:numColumns="3"  
 android:scrollbarAlwaysDrawHorizontalTrack="false"  
 android:scrollbarAlwaysDrawVerticalTrack="false"  
 android:verticalSpacing="2dp">  
 </GridView>  
 </LinearLayout>  
 </FrameLayout>  
 </HorizontalScrollView>  
  
</android.support.constraint.ConstraintLayout>

# PIELIKUMS 2

MainActivity.java

public class MainActivity extends AppCompatActivity {  
 static final int *REQUEST\_IMAGE\_CAPTURE* = 1;  
 String mCurrentPhotoPath;  
  
 private FirebaseAnalytics mFirebaseAnalytics;  
  
 private SectionsPageAdapter sectionsPageAdapter;  
 private ViewPager viewPager;  
  
 CameraFragment cameraFragment;  
 AudioFragment audioFragment;  
  
 public void logAnalyticsEvent(String event){  
 Bundle bundle = new Bundle();  
 bundle.putString(FirebaseAnalytics.Param.*ITEM\_NAME*, event);  
 mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.*SELECT\_CONTENT*, bundle);  
 }  
  
  
 private void setupViewPager(ViewPager viewPager){  
 SectionsPageAdapter adapter = new SectionsPageAdapter(getSupportFragmentManager());  
  
 cameraFragment = new CameraFragment();  
 audioFragment = new AudioFragment();  
  
 adapter.addFragment(cameraFragment, "Camera");  
 adapter.addFragment(audioFragment, "Audio");  
 viewPager.setAdapter(adapter);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.*toolbar*);  
 setSupportActionBar(toolbar);  
 // Obtain the FirebaseAnalytics instance.  
 mFirebaseAnalytics = FirebaseAnalytics.*getInstance*(this);  
  
 logAnalyticsEvent("launched application");  
  
 viewPager = (ViewPager) findViewById(R.id.*container*);  
 setupViewPager(viewPager);  
  
 TabLayout tabLayout = (TabLayout) findViewById(R.id.*tabs*);  
 tabLayout.setupWithViewPager(viewPager);  
  
  
// gridview.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
// public void onItemClick(AdapterView<?> parent, View v,  
// int position, long id) {  
// Toast.makeText(MainActivity.this, "" + position,  
// Toast.LENGTH\_SHORT).show();  
// }  
// });  
  
 }  
  
 @Override  
 */\*\*  
 \* Delete all images  
 \*/* protected void onDestroy() {  
 super.onDestroy();  
 File storageDir = getExternalFilesDir(Environment.*DIRECTORY\_PICTURES*);  
 logAnalyticsEvent("exit application");  
 this.*traverse*(storageDir);  
 }  
  
 public static void traverse (File dir) {  
 if (dir.exists()) {  
 File[] files = dir.listFiles();  
 for (int i = 0; i < files.length; ++i) {  
 File file = files[i];  
 if (file.isDirectory()) {  
 *traverse*(file);  
 } else {  
 // Delete files  
 file.delete();  
 }  
 }  
 }  
 }  
  
  
 public void recordAudio(View view){  
 logAnalyticsEvent("Record Audio");  
 audioFragment.toggleRecord();  
 }  
  
 */\*\*  
 \* Run a functiopn on button click, invoke devices camera  
 \** ***@param*** *view  
 \*/* public void catchImage(View view) {  
 logAnalyticsEvent("Catch image");  
  
 // Just show information that this is working, nothing more  
 Toast.*makeText*(this, "catch an image", Toast.*LENGTH\_LONG*).show();  
  
 // Invoke camera, available on device. Intent calls for available camera applications  
 Intent takePictureIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
  
 if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  
  
 // Create image filename  
 File photoFile = null;  
 try {  
 photoFile = createImageFile();  
 } catch (IOException ex) {  
 // Error occurred while creating the File  
  
 }  
  
 // File was created  
 if (photoFile != null) {  
 Uri photoURI = FileProvider.*getUriForFile*(this,  
 "com.teamtwo.md1.android.fileprovider",  
 photoFile);  
  
 takePictureIntent.putExtra(MediaStore.*EXTRA\_OUTPUT*, photoURI);  
 takePictureIntent.putExtra("data", photoURI);  
 startActivityForResult(takePictureIntent, *REQUEST\_IMAGE\_CAPTURE*);  
 }  
 }  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 if (requestCode == *REQUEST\_IMAGE\_CAPTURE* && resultCode == *RESULT\_OK*) {  
 Bitmap myBitmap = BitmapFactory.*decodeFile*(mCurrentPhotoPath);  
 cameraFragment.thumbnailView.setImageBitmap(myBitmap);  
 cameraFragment.adapter.addThisBitmap(myBitmap);  
 cameraFragment.adapter.notifyDataSetChanged();  
 }  
 }  
  
 */\*\*  
 \* Save image taken by camera inside external directory for every application that has permission to view images  
 \** ***@return*** *\** ***@throws*** *IOException  
 \*/* private File createImageFile() throws IOException {  
 // Create an image file name  
 String timeStamp = new SimpleDateFormat("yyyyMMdd\_HHmmss").format(new Date());  
 String imageFileName = "JPEG\_" + timeStamp + "\_";  
  
// File storageDir = getFilesDir();  
 File storageDir = getExternalFilesDir(Environment.*DIRECTORY\_PICTURES*);  
  
 File image = File.*createTempFile*(  
 imageFileName, /\* prefix \*/  
 ".jpg", /\* suffix \*/  
 storageDir /\* directory \*/  
 );  
  
 // Save a file: path for use with ACTION\_VIEW intents  
 mCurrentPhotoPath = image.getAbsolutePath();  
  
 return image;  
 }  
  
 public static class ImageAdapter extends BaseAdapter {  
 private Context mContext;  
 // References to our images  
 ArrayList<Bitmap> mThumbs = new ArrayList<Bitmap>();  
  
 public ImageAdapter(Context c) {  
 mContext = c;  
 }  
  
 public void addThisBitmap(Bitmap newmap) {  
 mThumbs.add(newmap);  
 }  
  
 public int getCount() {  
 return mThumbs.size();  
 }  
  
 public Object getItem(int position) {  
 return null;  
 }  
  
 public long getItemId(int position) {  
 return 0;  
 }  
  
 // create a new ImageView for each item referenced by the Adapter  
 public View getView(int position, View convertView, ViewGroup parent) {  
 ImageView imageView;  
 if (convertView == null) {  
 // if it's not recycled, initialize some attributes  
 imageView = new ImageView(mContext);  
 imageView.setLayoutParams(new ViewGroup.LayoutParams(500, 500));  
 imageView.setScaleType(ImageView.ScaleType.*CENTER\_CROP*);  
 imageView.setPadding(8, 8, 8, 8);  
 } else {  
 imageView = (ImageView) convertView;  
 }  
  
 imageView.setImageBitmap(mThumbs.get(position));  
 return imageView;  
 }  
 }  
}

# PIELIKUMS 3

AudioRecordTest.Java

public class AudioRecordTest extends AppCompatActivity {  
  
 private static final String *LOG\_TAG* = "AudioRecordTest";  
 private static final int *REQUEST\_RECORD\_AUDIO\_PERMISSION* = 200;  
 private static String *mFileName* = null;  
  
 private RecordButton mRecordButton = null;  
 private MediaRecorder mRecorder = null;  
  
 private PlayButton mPlayButton = null;  
 private MediaPlayer mPlayer = null;  
  
 // Requesting permission to RECORD\_AUDIO  
 private boolean permissionToRecordAccepted = false;  
 private String [] permissions = {Manifest.permission.*RECORD\_AUDIO*};  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 switch (requestCode){  
 case *REQUEST\_RECORD\_AUDIO\_PERMISSION*:  
 permissionToRecordAccepted = grantResults[0] == PackageManager.*PERMISSION\_GRANTED*;  
 break;  
 }  
 if (!permissionToRecordAccepted ) finish();  
  
 }  
  
 private void onRecord(boolean start) {  
 if (start) {  
 startRecording();  
 } else {  
 stopRecording();  
 }  
 }  
  
 private void onPlay(boolean start) {  
 if (start) {  
 startPlaying();  
 } else {  
 stopPlaying();  
 }  
 }  
  
 private void startPlaying() {  
 mPlayer = new MediaPlayer();  
 try {  
 mPlayer.setDataSource(*mFileName*);  
 mPlayer.prepare();  
 mPlayer.start();  
 } catch (IOException e) {  
 Log.*e*(*LOG\_TAG*, "prepare() failed");  
 }  
 }  
  
 private void stopPlaying() {  
 mPlayer.release();  
 mPlayer = null;  
 }  
  
 private void startRecording() {  
 mRecorder = new MediaRecorder();  
 mRecorder.setAudioSource(MediaRecorder.AudioSource.*MIC*);  
 mRecorder.setOutputFormat(MediaRecorder.OutputFormat.*THREE\_GPP*);  
 mRecorder.setOutputFile(*mFileName*);  
 mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.*AMR\_NB*);  
  
 try {  
 mRecorder.prepare();  
 } catch (IOException e) {  
 Log.*e*(*LOG\_TAG*, "prepare() failed");  
 }  
  
 mRecorder.start();  
 }  
  
 private void stopRecording() {  
 mRecorder.stop();  
 mRecorder.release();  
 mRecorder = null;  
 }  
  
 class RecordButton extends Button {  
 boolean mStartRecording = true;  
  
 OnClickListener clicker = new OnClickListener() {  
 public void onClick(View v) {  
 onRecord(mStartRecording);  
 if (mStartRecording) {  
 setText("Stop recording");  
 } else {  
 setText("Start recording");  
 }  
 mStartRecording = !mStartRecording;  
 }  
 };  
  
 public RecordButton(Context ctx) {  
 super(ctx);  
 setText("Start recording");  
 setOnClickListener(clicker);  
 }  
 }  
  
 class PlayButton extends Button {  
 boolean mStartPlaying = true;  
  
 OnClickListener clicker = new OnClickListener() {  
 public void onClick(View v) {  
 onPlay(mStartPlaying);  
 if (mStartPlaying) {  
 setText("Stop playing");  
 } else {  
 setText("Start playing");  
 }  
 mStartPlaying = !mStartPlaying;  
 }  
 };  
  
 public PlayButton(Context ctx) {  
 super(ctx);  
 setText("Start playing");  
 setOnClickListener(clicker);  
 }  
 }  
  
 @Override  
 public void onCreate(Bundle icicle) {  
 super.onCreate(icicle);  
  
 // Record to the external cache directory for visibility  
 *mFileName* = getExternalCacheDir().getAbsolutePath();  
 *mFileName* += "/audiorecordtest.3gp";  
  
 ActivityCompat.*requestPermissions*(this, permissions, *REQUEST\_RECORD\_AUDIO\_PERMISSION*);  
  
 LinearLayout ll = new LinearLayout(this);  
 mRecordButton = new RecordButton(this);  
 ll.addView(mRecordButton,  
 new LinearLayout.LayoutParams(  
 ViewGroup.LayoutParams.*WRAP\_CONTENT*,  
 ViewGroup.LayoutParams.*WRAP\_CONTENT*,  
 0));  
 mPlayButton = new PlayButton(this);  
 ll.addView(mPlayButton,  
 new LinearLayout.LayoutParams(  
 ViewGroup.LayoutParams.*WRAP\_CONTENT*,  
 ViewGroup.LayoutParams.*WRAP\_CONTENT*,  
 0));  
 setContentView(ll);  
 }  
  
 @Override  
 public void onStop() {  
 super.onStop();  
 if (mRecorder != null) {  
 mRecorder.release();  
 mRecorder = null;  
 }  
  
 if (mPlayer != null) {  
 mPlayer.release();  
 mPlayer = null;  
 }  
 }  
}

# PIELIKUMS 4

Fragment\_audio.xml

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout 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=".AudioFragment">  
  
  
<LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/recordAudioButton"  
 style="@android:style/Widget.DeviceDefault.Button.Borderless"  
 android:layout\_width="match\_parent"  
 android:layout\_height="47dp"  
 android:layout\_marginLeft="75dp"  
 android:layout\_marginRight="75dp"  
 android:background="@android:color/holo\_blue\_dark"  
 android:drawablePadding="8dp"  
 android:onClick="recordAudio"  
 android:text="Start recording"  
 tools:layout\_editor\_absoluteX="110dp"  
 tools:layout\_editor\_absoluteY="16dp"  
 tools:padding="8dp"  
 tools:text="Start recording" />  
  
 <TextView  
 android:id="@+id/currentRecordAudioName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:text=""  
 android:textColor="@android:color/black"  
 android:textSize="14sp"  
 tools:text="LOREM IPSUM" />  
  
  
 <ScrollView  
 android:layout\_width="368dp"  
 android:layout\_height="432dp"  
 tools:layout\_editor\_absoluteX="8dp"  
 tools:layout\_editor\_absoluteY="71dp">  
  
 <LinearLayout  
 android:id="@+id/audioButtonContainer"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 </LinearLayout>  
 </ScrollView>  
  
</LinearLayout>  
  
<!--<include layout="@layout/content\_fragment\_audio" />-->  
  
</FrameLayout>