|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** - **Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3** | |
| **по дисциплине** | |
| «Разработка программных приложений» | |
|  | |
| Выполнил студент группы ИКБО-02-18 | Юркевич Г.А. |
| Принял ассистент кафедры ИППО | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_2020 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. |  |

Москва 2020

**Содержание**

[1. Цель лабораторной работы 3](#_heading=h.26in1rg)

2. Общие требования [3](#_heading=h.2et92p0)

3. Задание [3](#_heading=h.3dy6vkm)

4. Поэтапное выполнение лабораторной работы [4](#_heading=h.4d34og8)

5. Результат работы [4](#_heading=h.lnxbz9)

6.Вывод [4](#_heading=h.17dp8vu)

[Список использованных источников 9](#_heading=h.35nkun2)

1. Цель лабораторной работы

Целью данной лабораторной работы является ознакомление со средой разработки программных приложений Android Studio, а также создание приложения из двух активити.

**2. Общие требования**

Среда разработки по умолчанию - Android Studio. Язык программирования - Java.

По своему желанию, можно использовать Kotlin, а также XCode/Swift при наличии соответствующего оборудования.

Решения должны быть размещены в виде проекта в git репозитории (github/bitbucket).

Ссылку на git-репозиторий и отчет по выполненной работе прислать на почту: strogankova-mirea@yandex.ru.

Минимальная версия API – 19/

**2. Задание**

Создать приложение, взаимодействующее с базой данных. Первое активити должно

содержать три кнопки. При нажатии на первую кнопку должно открываться новое

активити, выводящее информацию из таблицы Студенты в удобном для восприятия

формате.

При запуске приложения необходимо:

1

Создать БД, если ее не существует.

2

Создать таблицу Студенты, содержащую поля:

●

ID

●

ФИО

●

Время добавления записи

3

Удалять все записи из БД, а затем вносить 5 записей об​ ​одногруппниках со

случайными данными

При нажатии на вторую кнопку необходимо внести еще одну запись в таблицу.

При нажатии на третью кнопку необходимо заменить ФИО в последней внесенной

записи на Иванов Иван Иванович.

Создать также новое отдельное приложение на основе приложения, созданного в

части 1 Новое приложение должно иметь тот же самый package name, чтобы

обновлять первое при установке. Переопределить функцию onUpgrade. При

изменении изменить таблицу Студенты следующим образом:

●

ID

●

Фамилия

●

Имя

●

Отчество

●

Время добавления записи

Данные из поля ФИО необходимо по проблема разделить на три поля: Фамилия, Имя,

Отчество. Также, изменить версию базы данных

**4. Поэтапное выполнение лабораторной работы**

Создается база данных.

**5. Результат работы**

**MainActivity.java (вариант 2)**

package com.example.laba2

import android.os.AsyncTask

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.AdapterView

import kotlinx.android.synthetic.main.activity\_main.\*

import org.json.JSONArray

import java.net.URL

import android.content.Intent

import android.view.View

import android.widget.AdapterView.OnItemClickListener

import javax.net.ssl.HttpsURLConnection

class MainActivity : AppCompatActivity() {

var list = ArrayList<Tech>()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val url = "https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/data/techs.ruleset.json"

AsyncTaskHandleJson().execute(url)

technologies\_list.setOnItemClickListener(object : OnItemClickListener {

override fun onItemClick(parent: AdapterView<\*>, view: View, pos: Int, id: Long) {

val intent = Intent(view!!.context, VP::class.java)

var rPos = pos - 1

var lPos = pos + 1

if (pos == 0) rPos = list.lastIndex

if (pos == list.lastIndex) lPos = 0

intent.putExtra("1", list[rPos].graphic)

intent.putExtra("2", list[pos].graphic)

intent.putExtra("3", list[lPos].graphic)

intent.putExtra("4", list[rPos].name)

intent.putExtra("5", list[pos].name)

intent.putExtra("6", list[lPos].name)

intent.putExtra("7", list[rPos].helptext)

intent.putExtra("8", list[pos].helptext)

intent.putExtra("9", list[lPos].helptext)

startActivity(intent)

}

})

}

inner class AsyncTaskHandleJson : AsyncTask<String, String, String>() {

override fun doInBackground(vararg jsonUrl: String?): String {

val connection = URL(jsonUrl[0]).openConnection() as HttpsURLConnection

val jsonData: String

try{

connection.connect()

jsonData = connection.inputStream.use { it.reader().use{ reader -> reader.readText()} }

}finally {

connection.disconnect()

}

return jsonData

}

override fun onPostExecute(result: String?) {

super.onPostExecute(result)

HJson(result)

}

}

fun HJson(jsonString: String?) {

val jsonArray = JSONArray(jsonString)

var jsonLineIndex = 1

val adapter = ListAdapter(this, list)

while(jsonLineIndex < jsonArray.length())

{

val jsonObject = jsonArray.getJSONObject(jsonLineIndex)

list.add(

Tech(

jsonObject.getString("graphic"),

jsonObject.getString("name"),

if (jsonObject.has("helptext"))

jsonObject.getString("helptext")

else

""

)

)

jsonLineIndex++

}

technologies\_list.adapter = adapter

}

}

**MainActivity.java (вариант 1)**

package com.example.android\_lab3;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Random;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

import android.provider.BaseColumns;

import android.view.View;

import android.widget.Button;

import android.content.ContentValues;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

Button viewBDButton,insRecButton,upLastRecButton;

StudentDBHelper dbHelper;

public int count=5;

public String[] strArray1= new String[] {"Галкин","Одинцов","Колобов","Кузнецов","Агафонов","Пестов","Николаев","Артемьев","Козлов","Князев"};

public String[] strArray2= new String[] {" Вилли"," Гордей"," Мирослав"," Осип"," Даниил"," Захар"," Елисей"," Борис"," Марк"," Савелий"};

public String[] strArray3= new String[] {" Якунович"," Валентинович"," Антонинович"," Вячеславович"," Вадимович"," Богуславович"," Авдеевич"," Андреевич"," Анатольевич"," Борисович"};

Date currentDate = new Date();

DateFormat timeFormat = new SimpleDateFormat("HH:mm:ss", Locale.getDefault());

String timeText = timeFormat.format(currentDate);

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.acivity\_main);

dbHelper= new StudentDBHelper(this);

final SQLiteDatabase database = dbHelper.getWritableDatabase();

final ContentValues contentValues = new ContentValues();

View.OnClickListener oclBtn1 = new View.OnClickListener() {

@Override

public void onClick(View v) {

Cursor cursor = database.query(StudentDBHelper.TABLE\_CONTACTS, null, null, null, null, null, null);

ArrayList BD = new ArrayList();

if (cursor.moveToFirst()) {

int idIndex = cursor.getColumnIndex(StudentDBHelper.KEY\_ID);

int nameIndex = cursor.getColumnIndex(StudentDBHelper.FIO);

int dateIndex = cursor.getColumnIndex(StudentDBHelper.dateOfAdd);

do {

BD.add("ID:" + cursor.getInt(idIndex)+" ФИО:" + cursor.getString(nameIndex) +" время:"+ cursor.getString(dateIndex));

Log.d("mLog", "ID = " + cursor.getInt(idIndex) +

", name = " + cursor.getString(nameIndex) +

", email = " + cursor.getString(dateIndex));

} while (cursor.moveToNext());

} else

Log.d("mLog","0 rows");

cursor.close();

Intent intent = new Intent(getApplicationContext(),LIst.class);

intent.putExtra("list", BD);

startActivity(intent);

}

};

View.OnClickListener oclBtn2 = new View.OnClickListener() {

@Override

public void onClick(View v) {

currentDate = new Date();

timeFormat = new SimpleDateFormat("HH:mm:ss", Locale.getDefault());

timeText = timeFormat.format(currentDate);

Random random = new Random();

int num1 = random.nextInt(11);

int num2 = random.nextInt(11);

int num3 = random.nextInt(11);

contentValues.put(StudentDBHelper.FIO, (strArray1[num1] + strArray2[num2] + strArray3[num3]));

contentValues.put(StudentDBHelper.dateOfAdd,timeText);

database.insert(StudentDBHelper.TABLE\_CONTACTS, null, contentValues);

count+=1;

}

};

View.OnClickListener oclBtn3 = new View.OnClickListener() {

@Override

public void onClick(View v) {

currentDate = new Date();

timeFormat = new SimpleDateFormat("HH:mm:ss", Locale.getDefault());

timeText = timeFormat.format(currentDate);

ContentValues cv=new ContentValues();

cv.put(StudentDBHelper.FIO, "Иванов Иван Иванович");

cv.put(StudentDBHelper.dateOfAdd, timeText);

database.update(StudentDBHelper.TABLE\_CONTACTS,cv,"\_id = ?",new String[] { (Integer.toString(count)) });

}

};

viewBDButton= (Button) findViewById(R.id.viewBDButton);

viewBDButton.setOnClickListener(oclBtn1);

insRecButton= (Button) findViewById(R.id.insRecButton);

insRecButton.setOnClickListener(oclBtn2);

upLastRecButton= (Button) findViewById(R.id.updLastRecButton);

upLastRecButton.setOnClickListener(oclBtn3);

database.delete(StudentDBHelper.TABLE\_CONTACTS, null, null);

for (int x = 0; x < 5; x = x + 1) {

Random random = new Random();

int num1 = random.nextInt(11);

int num2 = random.nextInt(11);

int num3 = random.nextInt(11);

contentValues.put(StudentDBHelper.FIO, (strArray1[num1] + strArray2[num2] + strArray3[num3]));

contentValues.put(StudentDBHelper.dateOfAdd,timeText);

database.insert(StudentDBHelper.TABLE\_CONTACTS, null, contentValues);

}

}

}

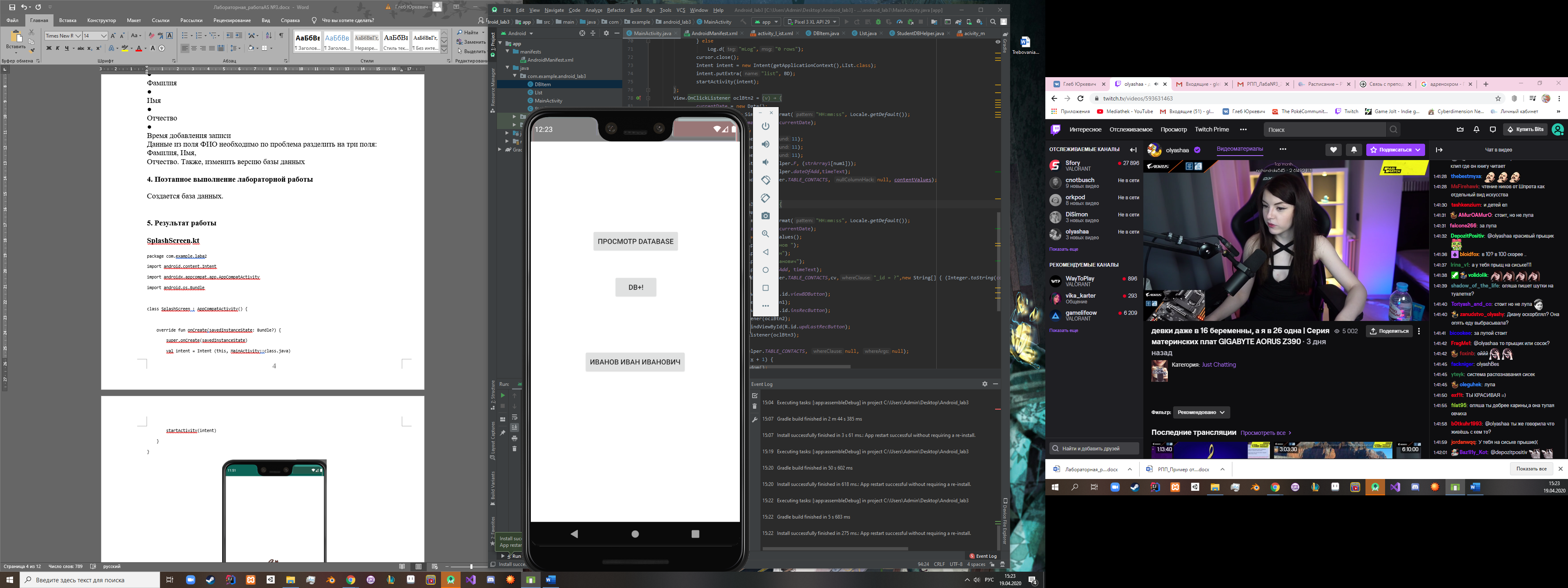


Рисунок 1 Первое активити

**list.java**

package com.example.android\_lab3;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import java.util.ArrayList;

public class LIst extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_l\_ist);

ArrayList<String> arrayFromIntent = (ArrayList<String>) getIntent().getSerializableExtra("list");

ListView list = (ListView) findViewById(R.id.list);

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

android.R.layout.simple\_list\_item\_1, arrayFromIntent);

list.setAdapter(adapter);

}

}

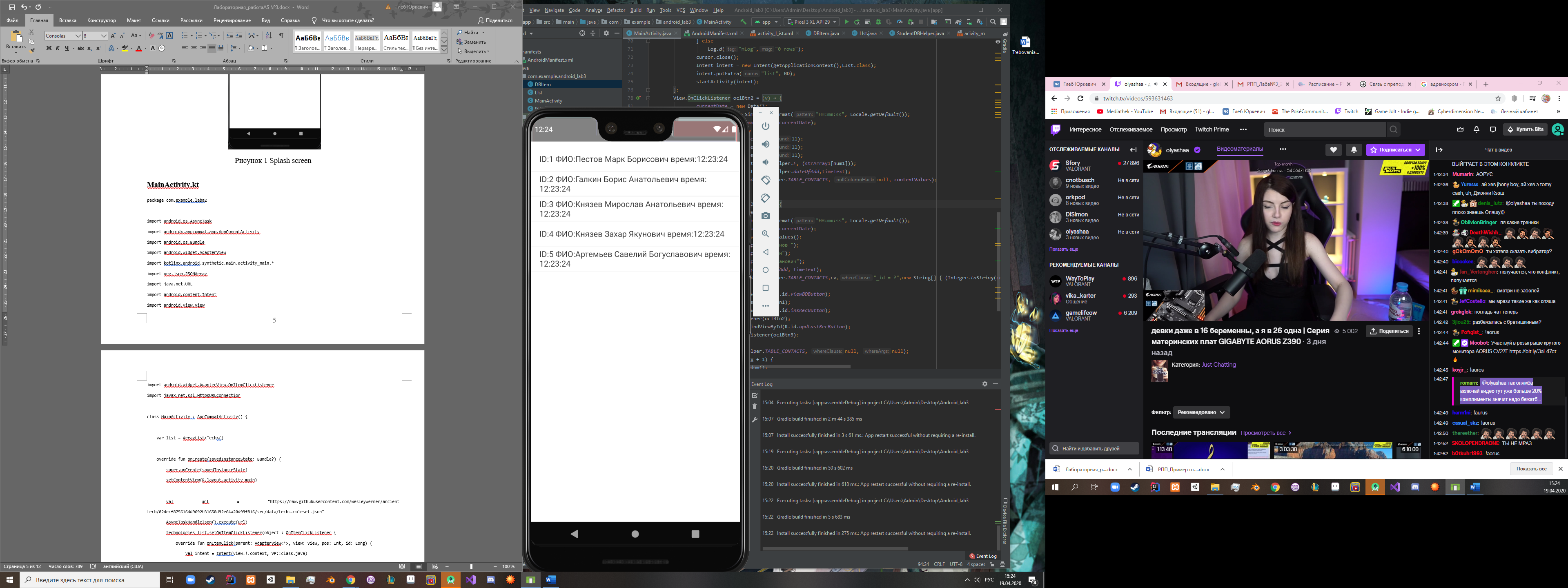


Рисунок 2 ListView

**StudentDBHelper.java (версия 2)**

package com.example.android\_lab3;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.os.Bundle;

import androidx.annotation.Nullable;

public class StudentDBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 2;

public static final String DATABASE\_NAME = "StudentDB";

public static final String TABLE\_CONTACTS = "Student";

public static final String KEY\_ID = "\_id";

public static final String F = "F";

public static final String I = "I";

public static final String O = "O";

public static final String dateOfAdd = "date";

public StudentDBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + TABLE\_CONTACTS + "(" + KEY\_ID + " integer primary key," + F + " text," + O + " text,"+ I + " text,"+ dateOfAdd + " text" + ")");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("drop table if exists " + TABLE\_CONTACTS);

onCreate(db);

}

}

**StudentDBHelper.java (версия 1)**

package com.example.android\_lab3;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.os.Bundle;

import androidx.annotation.Nullable;

public class StudentDBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 1;

public static final String DATABASE\_NAME = "StudentDB";

public static final String TABLE\_CONTACTS = "Student";

public static final String KEY\_ID = "\_id";

public static final String FIO = "FIODB";

public static final String dateOfAdd = "date";

public StudentDBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + TABLE\_CONTACTS + "(" + KEY\_ID + " integer primary key," + FIO + " text," + dateOfAdd + " text" + ")");

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("drop table if exists " + TABLE\_CONTACTS);

onCreate(db);

}

}

**5. Вывод**

В данной работе было скачано, установлено и запущено приложение Android Studio. Также было реализовано приложение, состоящее из двух активити использующее базу данных. Приложение было протестировано.

**Список использованных источников**

1. Главный сайт по Android: <https://developer.android.com/>
2. Канал Google на YouTube:

<http://www.youtube.com/channel/UC_x5XG1OV2P6uZZ5FSM9Ttw>

1. Онлайн-курсы по Android:

– Udacity:

<https://eu.udacity.com/course/new-android-fundamentals--ud851>

1. <https://stackoverflow.com/>
2. <https://github.com>
3. <https://google.com/>