|  |
| --- |
| 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 №4** | |
| **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/

**3. Задание**

Создать приложение, взаимодействующее с базой данных. Первое активити должно

Разработать клиент для сервиса​ t​hecatapi.com. ​Сервис выдает картинки с кошками и  
имеет возможность поиска по породе. Приложение должно иметь форму для выбора  
породы, кнопку поиска и компоненту для отображения результата.  
Требования по функционалу:  
1.Сохранять последние параметры формы.  
2.Реализовать бесконечный скроллинг.  
3.Реализовать возможность голосования за картинку (кнопки лайк/дизлайк).  
4.Запоминать последние 10 лайков и сделать отдельный экран для их  
отображения

**4. Поэтапное выполнение лабораторной работы**

Создается первое активити, работающее как меню для приложения, после чего создается второе и третье активити представляющие из себя прокручиваемые списки. Для каждого списка были созданы адаптеры. Элементы первого листа состоят из картинки и двух кнопок с надписями «Love it» и «Nope it». Элементы второго листа это последние десять картинок, для которых была нажата кнопка с надписью «Love it» в первом списке. Была реализована возможность поиска по породам, а также, сохранение последних параметров формы.

**5. Результат работы**

**MainActivity**

package com.example.myclientserver;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.content.ContentValues;

import android.content.Intent;

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;

import android.widget.Toast;

import com.android.volley.RequestQueue;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.JsonArrayRequest;

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.HttpUrl;

public class MainActivity extends AppCompatActivity {

private boolean loading = true;

int pastVisibleItems, visibleItemCount, totalItemCount;

private RecyclerView mRecyclerView;

private ExampleAdapter mExampleAdapter;

private JsonArrayRequest requests;

DBHelper dataBaseHelper;

Button b1;

Button b2;

EditText ed;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

dataBaseHelper = new DBHelper(this);

ed=(EditText) findViewById(R.id.editText3) ;

b1=(Button) findViewById(R.id.ok) ;

b2=(Button) findViewById(R.id.ten) ;

final LinearLayoutManager mLayoutManager;

mLayoutManager = new LinearLayoutManager(this);

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

@Override

public void onClick(View v) {

String str= ed.getText().toString();

String breed = "";

switch (str) {

case "":

HttpUrl.Builder urlBuilder = HttpUrl.parse("https://api.thecatapi.com/v1/images/search").newBuilder();

urlBuilder.addQueryParameter("limit", "100");

urlBuilder.addQueryParameter("page", "10");

urlBuilder.addQueryParameter("order", "Desc");

urlBuilder.addQueryParameter("mime\_types", "gif,jpg,png");

String url = urlBuilder.build().toString();

Intent intent1 = new Intent(MainActivity.this,ListActivity.class);

intent1.putExtra("URL", url);

startActivity(intent1);

break;

case "abyssianian":

breed = "abys";

breedUrl(breed);

break;

case "aegean":

breed = "aege";

breedUrl(breed);

break;

case "balinese":

breed = "bali";

breedUrl(breed);

break;

case "bengal":

breed = "beng";

breedUrl(breed);

break;

case "burmese":

breed = "burm";

breedUrl(breed);

break;

case "chartreux":

breed = "char";

breedUrl(breed);

break;

case "cymric":

breed = "cymr";

breedUrl(breed);

break;

case "persian":

breed = "pers";

breedUrl(breed);

break;

case "ragdoll":

breed = "ragd";

breedUrl(breed);

break;

case "savannah":

breed = "sava";

breedUrl(breed);

break;

case "sphynx":

breed = "sphy";

breedUrl(breed);

break;

case "toyger":

breed = "toyg";

breedUrl(breed);

break;

}

}

};

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

@Override

public void onClick(View v) {

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

startActivity(intent);

finish();

}

};

b1.setOnClickListener(oclBtn1);

b2.setOnClickListener(oclBtn2);

}

public void breedUrl(String breed) {

HttpUrl.Builder urlBuilder = HttpUrl.parse("https://api.thecatapi.com/v1/images/search").newBuilder();

urlBuilder.addQueryParameter("breed\_id", breed);

urlBuilder.addQueryParameter("limit", "100");

urlBuilder.addQueryParameter("page", "10");

urlBuilder.addQueryParameter("order", "Desc");

urlBuilder.addQueryParameter("mime\_types", "gif,jpg,png");

String url = urlBuilder.build().toString();

Intent intent = new Intent(MainActivity.this,ListActivity.class);

intent.putExtra("URL", url);

startActivity(intent);

}

public void buttonDislike(View view) {

Toast.makeText(this, "Вам не понравилось", Toast.LENGTH\_SHORT).show();

}

public void buttonLike(View view) {

Toast.makeText(this, "Вам понравилось", Toast.LENGTH\_SHORT).show();

}

}

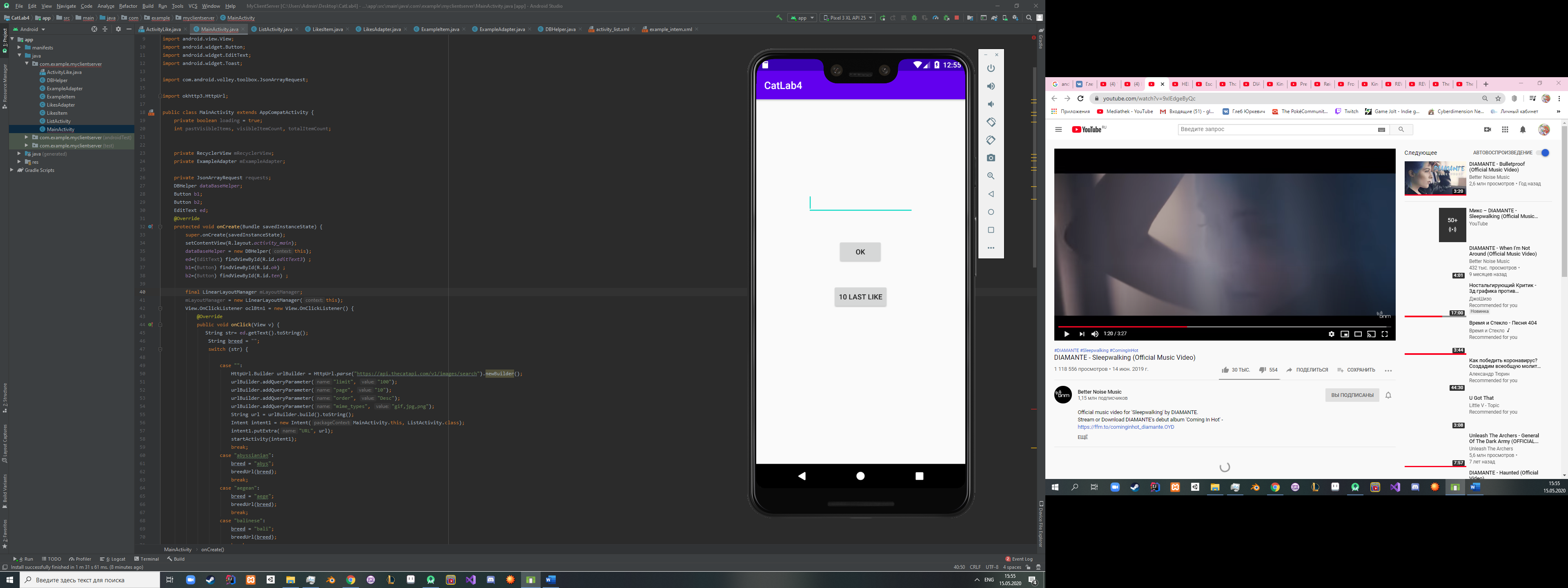


Рисунок 1. Первое активити

**ListActivity**

package com.example.myclientserver;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.content.ContentValues;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Toast;

import com.android.volley.RequestQueue;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.JsonArrayRequest;

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.util.ArrayList;

import okhttp3.HttpUrl;

public class ListActivity extends AppCompatActivity implements ExampleAdapter.OnItemClickListener {

private ArrayList<ExampleItem> mExampleList;

private RequestQueue mRequestQueue;

private JsonArrayRequest requests;

private ExampleAdapter mExampleAdapter;

private RecyclerView mRecyclerView;

DBHelper dataBaseHelper;

private boolean loading = true;

int pastVisiblesItems, visibleItemCount, totalItemCount;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_list);

String url= getIntent().getStringExtra("URL");

mExampleList = new ArrayList<>();

mRequestQueue = Volley.newRequestQueue(this);

mExampleList.clear();

dataBaseHelper = new DBHelper(this);

mRecyclerView = findViewById(R.id.recycler\_view);

mRecyclerView.setHasFixedSize(true);

final LinearLayoutManager mLayoutManager;

mLayoutManager = new LinearLayoutManager(this);

mRecyclerView.setLayoutManager(mLayoutManager);

mExampleList = new ArrayList<>();

mRequestQueue = Volley.newRequestQueue(this);

mRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

if (dy > 0) { //check for scroll down

visibleItemCount = mLayoutManager.getChildCount();

totalItemCount = mLayoutManager.getItemCount();

pastVisiblesItems = mLayoutManager.findFirstVisibleItemPosition();

if (loading) {

if ((visibleItemCount + pastVisiblesItems) >= totalItemCount) {

loading = false;

Log.v("...", "Last Item Wow !");

HttpUrl.Builder urlBuilder = HttpUrl.parse("https://api.thecatapi.com/v1/images/search").newBuilder();

urlBuilder.addQueryParameter("limit", "100");

urlBuilder.addQueryParameter("page", "10");

urlBuilder.addQueryParameter("order", "Desc");

urlBuilder.addQueryParameter("mime\_types", "gif,jpg,png");

String url = urlBuilder.build().toString();

parseJSON(url);

}

}

}

}

});

requests = new JsonArrayRequest(url, new com.android.volley.Response.Listener<JSONArray>() {

@Override

public void onResponse(JSONArray response) {

JSONObject hit = null;

for (int i = 1; i < response.length(); i++) {

try {

hit = response.getJSONObject(i);

String imageUrl = hit.getString("url");

mExampleList.add(new ExampleItem(imageUrl));

mExampleAdapter = new ExampleAdapter(ListActivity.this, mExampleList);

mRecyclerView.setAdapter(mExampleAdapter);

mExampleAdapter.setOnItemClickListener(ListActivity.this);

} catch (JSONException e) {

e.printStackTrace();

}

}

}

}, new com.android.volley.Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

error.printStackTrace();

}

});

mRequestQueue.add(requests);

}

public void buttonDislike(View view) {

Toast.makeText(this, "Вам не понравилось", Toast.LENGTH\_SHORT).show();

}

public void buttonLike(View view) {

Toast.makeText(this, "Вам понравилось", Toast.LENGTH\_SHORT).show();

}

@Override

public void onItemClick(int position) {

ExampleItem clickedItem = mExampleList.get(position);

String url =clickedItem.getImageUrl();

final SQLiteDatabase database = dataBaseHelper.getWritableDatabase();

final ContentValues contentValues = new ContentValues();

contentValues.put(DBHelper.URL,(url));

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

}

public void parseJSON(String url) {

mExampleList.clear();

requests = new JsonArrayRequest(url, new com.android.volley.Response.Listener<JSONArray>() {

@Override

public void onResponse(JSONArray response) {

JSONObject hit = null;

for (int i = 1; i < response.length(); i++) {

try {

hit = response.getJSONObject(i);

String imageUrl = hit.getString("url");

mExampleList.add(new ExampleItem(imageUrl));

mExampleAdapter = new ExampleAdapter(ListActivity.this, mExampleList);

mRecyclerView.setAdapter(mExampleAdapter);

mExampleAdapter.setOnItemClickListener(ListActivity.this);

} catch (JSONException e) {

e.printStackTrace();

}

}

}

}, new com.android.volley.Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

error.printStackTrace();

}

});

}}

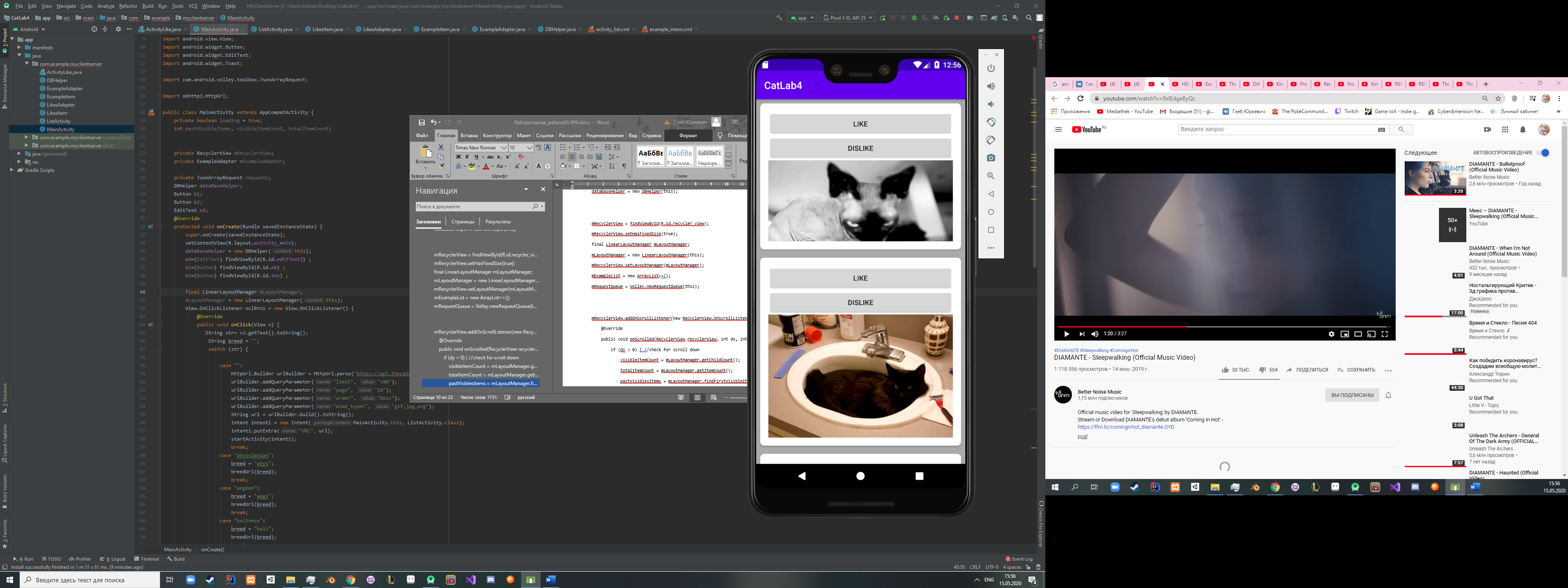


Рисунок 2. Лист 1

**LikesAdapter**

package com.example.myclientserver;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageView;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.ArrayList;

public class LikesAdapter extends RecyclerView.Adapter<LikesAdapter.ExampleViewHolder> {

private Context mContext;

private ArrayList<LikesItem> mExampleList;

public LikesAdapter(Context context, ArrayList<LikesItem> exampleList) {

mContext = context;

mExampleList = exampleList;

}

@Override

public ExampleViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View v = LayoutInflater.from(mContext).inflate(R.layout.likes\_intem, parent, false);

return new ExampleViewHolder(v);

}

@Override

public void onBindViewHolder(ExampleViewHolder holder, int position) {

LikesItem currentItem = mExampleList.get(position);

String imageUrl = currentItem.getImageUrl();

Glide.with(mContext)

.load(imageUrl)

.placeholder(R.drawable.unnamed)

.fitCenter()

.into(holder.mImageView);

}

@Override

public int getItemCount() {

return mExampleList.size();

}

public class ExampleViewHolder extends RecyclerView.ViewHolder {

public ImageView mImageView;

public ExampleViewHolder(View itemView) {

super(itemView);

mImageView = itemView.findViewById(R.id.image\_views);

}

}

}

**ExampleAdapter**

package com.example.myclientserver;

import android.content.Context;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.ImageView;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.ArrayList;

public class ExampleAdapter extends RecyclerView.Adapter<ExampleAdapter.ExampleViewHolder> {

private Context mContext;

private ArrayList<ExampleItem> mExampleList;

private OnItemClickListener mListener;

public interface OnItemClickListener {

void onItemClick(int position);

}

public void setOnItemClickListener(OnItemClickListener listener) {

mListener = listener;

}

public ExampleAdapter(Context context, ArrayList<ExampleItem> exampleList) {

mContext = context;

mExampleList = exampleList;

}

@Override

public ExampleViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

View v = LayoutInflater.from(mContext).inflate(R.layout.example\_intem, parent, false);

return new ExampleViewHolder(v);

}

@Override

public void onBindViewHolder(ExampleViewHolder holder, int position) {

ExampleItem currentItem = mExampleList.get(position);

String imageUrl = currentItem.getImageUrl();

Glide.with(mContext)

.load(imageUrl)

.placeholder(R.drawable.unnamed)

.fitCenter()

.into(holder.mImageView);

if ((position >= getItemCount() - 1)) {

load();

}

}

@Override

public int getItemCount() {

return mExampleList.size();

}

public void load(){

}

public class ExampleViewHolder extends RecyclerView.ViewHolder {

public ImageView mImageView;

public Button button;

public Button button1;

public ExampleViewHolder(View itemView) {

super(itemView);

button = itemView.findViewById(R.id.like);

mImageView = itemView.findViewById(R.id.image\_view);

button1 = itemView.findViewById(R.id.dislike);

itemView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mListener != null) {

int position = getAdapterPosition();

if (position != RecyclerView.NO\_POSITION) {

mListener.onItemClick(position);

}

}

}

});

}

}

}

**DBHelper**

package com.example.myclientserver;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

public class DBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 1;

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

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

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

public static final String URL = "Url";

public DBHelper(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," + URL + " text" + ")");

}

@Override

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

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

onCreate(db);

}

}

**ActivityLike**

package com.example.myclientserver;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import java.util.ArrayList;

public class ActivityLike extends AppCompatActivity {

private RecyclerView mRecyclerView;

private LikesAdapter mExampleAdapter;

private ArrayList<LikesItem> mExampleList;

DBHelper dbHelper;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_like);

mRecyclerView = findViewById(R.id.recycler\_views);

mRecyclerView.setHasFixedSize(true);

mRecyclerView.setLayoutManager(new LinearLayoutManager(this));

mExampleList = new ArrayList<>();

dbHelper = new DBHelper(this);

final SQLiteDatabase database = dbHelper.getWritableDatabase();

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

ArrayList<String> BD = new ArrayList();

if (cursor.moveToFirst()) {

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

int nameIndex = cursor.getColumnIndex(DBHelper.URL);

do {

BD.add(cursor.getString(nameIndex));

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

", URL = " + cursor.getString(nameIndex));

} while (cursor.moveToNext());

} else

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

cursor.close();

parse(BD);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_select, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.cats:

Intent intent = new Intent(this, MainActivity.class);

startActivity(intent);

finish();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

public void parse(ArrayList<String> BD) {

mExampleList.clear();

int size = BD.size();

if (size <= 10) {

for (int i = 0; i < BD.size(); i++) {

String catUrl = BD.get(i);

mExampleList.add(new LikesItem(catUrl));

mExampleAdapter = new LikesAdapter(ActivityLike.this, mExampleList);

mRecyclerView.setAdapter(mExampleAdapter);

}

} else {

int length = BD.size() - 10;

for (int i = length; i < BD.size(); i++) {

String catUrl = BD.get(i);

mExampleList.add(new LikesItem(catUrl));

mExampleAdapter = new LikesAdapter(ActivityLike.this, mExampleList);

mRecyclerView.setAdapter(mExampleAdapter);

}

}

}

}

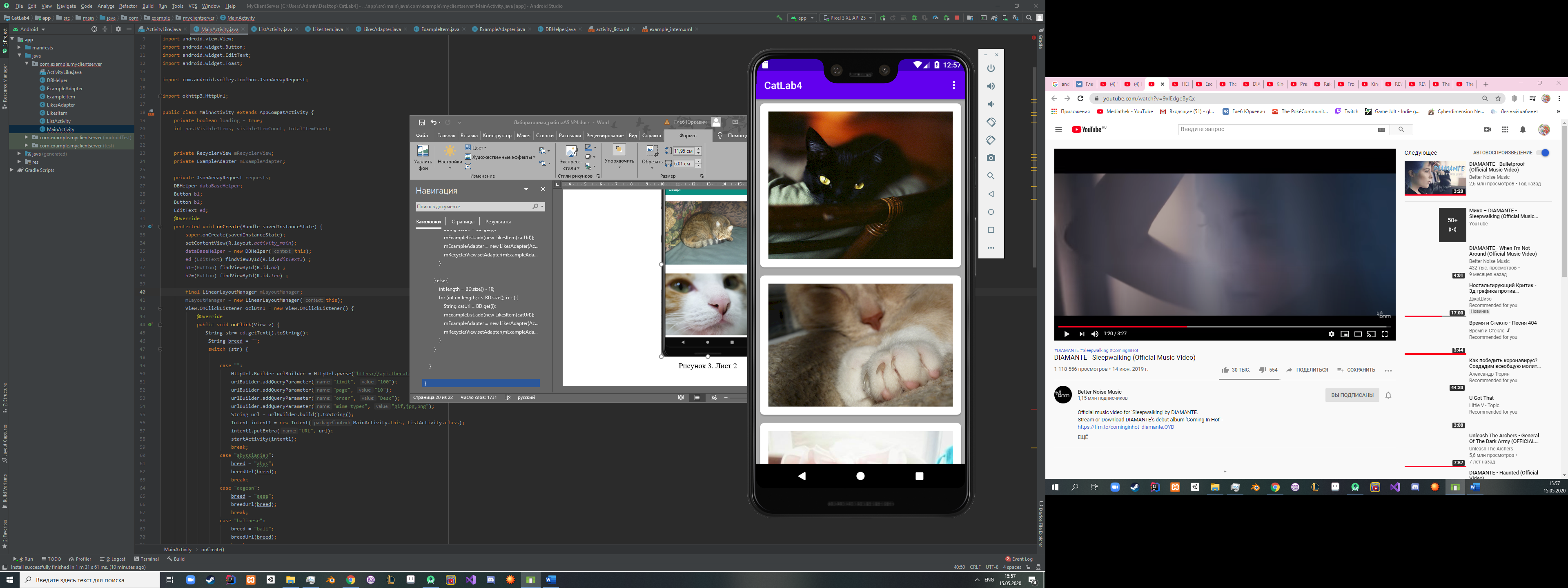


Рисунок 3. Лист 2

**5. Вывод**

В данной работе был разработан клиент для сервиса​ t​hecatapi.com. ​Приложение выдает картинки с кошками и имеет возможность поиска по породе, позволяет ставить лайк/дизлайк, а также сохраняет последние 10 картинок, которым был поставлен лайк и позволяет их просматривать.

**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/>