



WYŻSZA SZKOŁA INFORMATYKI i ZARZĄDZANIA z siedzibą w Rzeszowie

Dokumentacja z projektu.

Prowadzący: Pan Marcin Ochyra

Mikołaj Mryczko w57020

Maciej Głuszyk w57004

4 IID

Rzeszów, 2018

1. Opis aplikacji

Głównym celem stworzonej przez nas aplikacji jest, tworzenie listy zakupów. Cała Aplikacja będzie składać się z 4 głównych modułów:

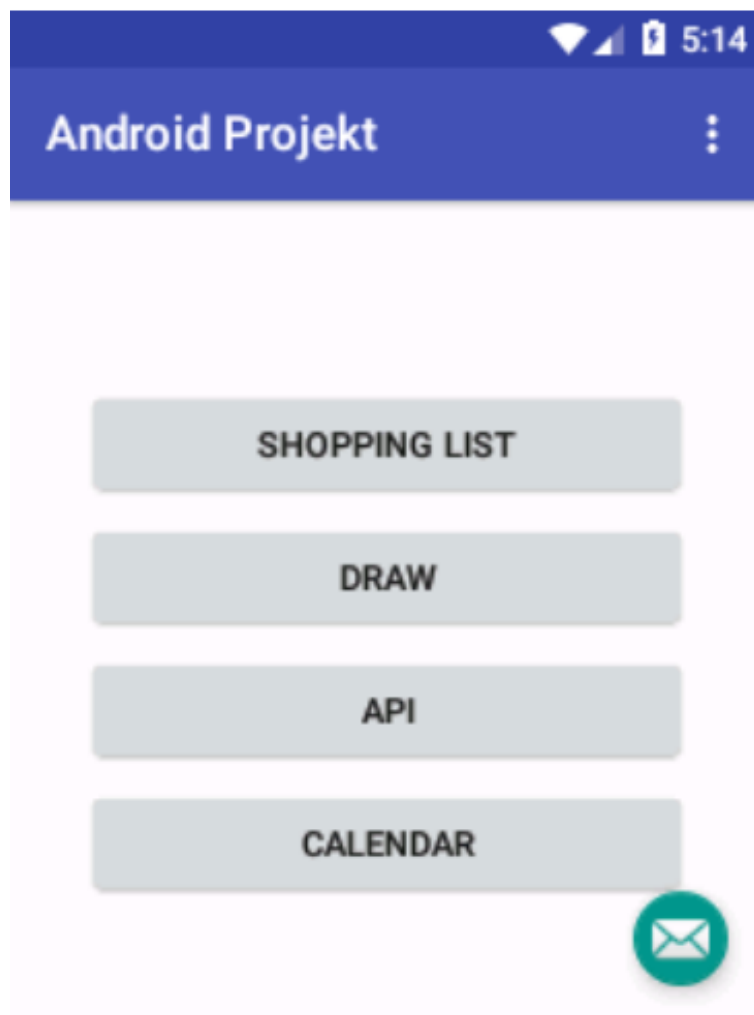
- Tworzenie listy zakupów, oraz usuwanie poszczególnych elementów w dowolnym momencie.
- Aplikacja która pozwala na tworzenie rysunków.
- Pobieranie danych z zewnętrznego api i json.
- Własny widok, który pozwala na zapisanie tekstu lub zmianę wartości.

W aplikacji użyliśmy: list, oraz adapterów do wyświetlenia wyników. W widokach również zauważalne są CheckBox'y.

2. Główny Widok

Głównym widokiem w aplikacji jest menu w którym wybieramy co chcemy robić.

Android Emulator - 2.7_QVGA_API_25:5554



Kod który znajduje się w layoutie tego widoku :

```
<Button
    android:id="@+id/shoppingListButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Shopping list"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="32dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintVertical_bias="0.25"
    android:layout_marginRight="32dp"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintHorizontal_bias="0.497" />

<Button
    android:id="@+id/drawButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="8dp"
    android:text="Draw"
    app:layout_constraintLeft_toLeftOf="@+id/shoppingListButton"
    app:layout_constraintRight_toRightOf="@+id/shoppingListButton"
    app:layout_constraintTop_toBottomOf="@+id/shoppingListButton" />
```

```
<Button
    android:id="@+id/apiButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="8dp"
    android:text="API"
    app:layout_constraintLeft_toLeftOf="@+id/drawButton"
    app:layout_constraintRight_toRightOf="@+id/drawButton"
    app:layout_constraintTop_toBottomOf="@+id/drawButton" />

<Button
    android:id="@+id/viewButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="8dp"
    android:text="Calendar"
    app:layout_constraintLeft_toLeftOf="@+id/apiButton"
    app:layout_constraintRight_toRightOf="@+id/apiButton"
    app:layout_constraintTop_toBottomOf="@+id/apiButton" />
```

W każdym przycisku wyznaczamy jego położenie w layoucie, dodatkowo przypisujemy nazwę przycisku, każdy taki przycisk swoją nazwę zapisuje w pliku String.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name"> Android Projekt</string>
    <string name="action_info">Projekt aplikacje mobilne</string>
    <string name="shopping_list_button">Shopping list</string>
    <string name="draw_button">Draw</string>
    <string name="api_button">API</string>
    <string name="view_button">Calendar</string>
    <string name="html_button">Data from HTML</string>
    <string name="json_button">Data from JSON</string>
    <string name="title_activity_shopping_list">Shopping list</string>
    <string name="course_info">Projekt</string>
    <string name="dialog_message">Projekt aplikacje mobilne. </string>
    <string name="dialog_title">Info</string>
    <string name="ok">OK</string>
</resources>
```

W klasie MainActivity znajdują się metody które pozwalają na działanie tych przycisków, kiedy naciskamy odpowiedni przycisk zostajemy przeniesieni w dalszą część aplikacji, oraz są zawarte wszystkie czynności jaka ta aplikacja wykonuje.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ButterKnife.bind(target: this);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
}

@OnClick(R.id.shoppingListButton)
void onClickShoppingList() {
    Intent intent = new Intent(packageContext: this, ShoppingListActivity.class);
    startActivity(intent);
}

@OnClick(R.id.drawButton)
void onClickDraw() {
    Intent intent = new Intent(packageContext: this, DrawActivity.class);
    startActivity(intent);
}

@OnClick(R.id.apiButton)
void onClickApi() {
    Intent intent = new Intent(packageContext: this, ApiActivity.class);
    startActivity(intent);
}

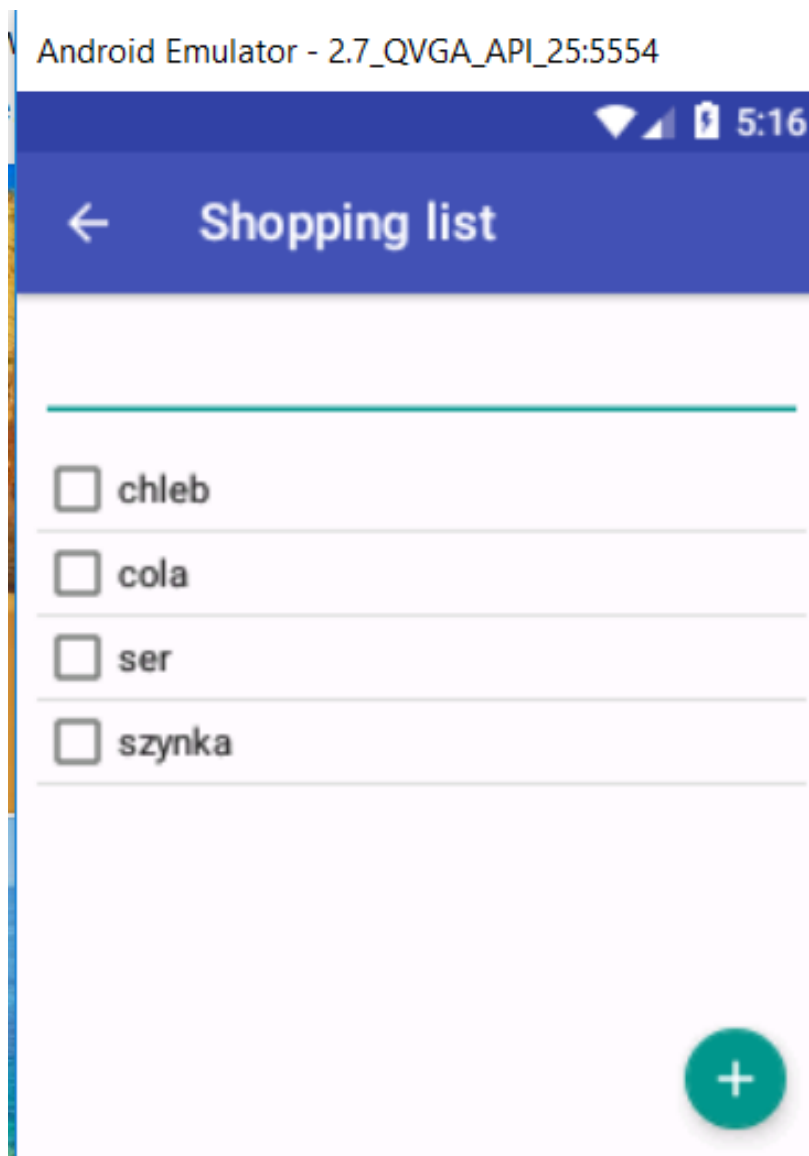
@OnClick(R.id.viewButton)
void onClickView() {
    Intent intent = new Intent(packageContext: this, ViewActivity.class);
    startActivity(intent);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    return super.onOptionsItemSelected(item);
}
```

3.Lista zakupów

Jest to główna idea tego programu. Dodajemy produkty pisząc nazwy w przeznaczonym na to miejscu. Produkty dodają się do listy, po naciśnięciu checkboxa znikają z niej. W tym widoku znajdują się Backward Compatibility, czyli możliwość powrotu do głównego menu.



W layaoucie znajdują się 3 elementy: EditText, ListView, oraz Spinner. Wszystkie są powiązane ze sobą w widoku.

```
<EditText
    android:id="@+id/itemName_ET"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<ListView
    android:id="@+id/itemList"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/itemName_ET"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toTopOf="@+id/guideline2" />

<Spinner
    android:id="@+id/itemSpinner"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:spinnerMode="dialog"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline2" />
```

W klasie ShoppingListActivity są zapisane wszystkie metody które pozwalają na dodanie nowych elementów do listy oraz ich zapisanie. Również zasady działania Spinnerów i checkboxów . Cały kod jest rozbudowany.

```
@BindView(R.id.itemName_ET)
EditText itemName;
@BindView(R.id.itemList)
ListView itemList;
@BindView(R.id.itemSpinner)
Spinner itemSpinner;
private List<String> listItems;
private List<String> spinnerItems;
private Set<String> listSetItems;
private Set<String> spinnerSetItems;
private static final String LIST_ITEMS_KEY = "LIST_ITEMS_KEY";
private static final String SPINNER_ITEMS_KEY = "SPINNER_ITEMS_KEY";
private static final String SHOPPING_LIST_KEY = "SHOPPING_LIST_KEY";
private ShoppingListAdapter listAdapter;
private ArrayAdapter<String> spinnerAdapter;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_shopping_list);
    ButterKnife.bind(target this);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener((view) -> {
        if (itemName.getText() != null && !itemName.getText().toString().trim().isEmpty() && !listSetItems.add(itemName.getText().toString())) {
            listItems.add(itemName.getText().toString());
            itemName.setText("");
            listAdapter.notifyDataSetChanged();
        }
    });
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    SharedPreferences sp = getSharedPreferences(SHOPPING_LIST_KEY, MODE_PRIVATE);
    listSetItems = sp.getStringSet(LIST_ITEMS_KEY, new HashSet<String>());
    spinnerSetItems = sp.getStringSet(SPINNER_ITEMS_KEY, new HashSet<String>());
    listItems = new ArrayList<>(listSetItems);
    spinnerItems = new ArrayList<>(spinnerSetItems);
    spinnerAdapter = new ArrayAdapter<>(context this, android.R.layout.select_dialog_item, spinnerItems);
    itemSpinner.setAdapter(spinnerAdapter);
    itemSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
```

```
});
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
SharedPreferences sp = getSharedPreferences(SHOPPING_LIST_KEY, MODE_PRIVATE);
listSetItems = sp.getStringSet(LIST_ITEMS_KEY, new HashSet<String>());
spinnerSetItems = sp.getStringSet(SPINNER_ITEMS_KEY, new HashSet<String>());
listItems = new ArrayList<>(listSetItems);
spinnerItems = new ArrayList<>(spinnerSetItems);
spinnerAdapter = new ArrayAdapter<>(context this, android.R.layout.select_dialog_item, spinnerItems);
itemSpinner.setAdapter(spinnerAdapter);
itemSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        if (!spinnerItems.get(position).equals("")) {
            itemName.setText("" + spinnerItems.get(position));
            spinnerItems.remove(position);
            spinnerAdapter.notifyDataSetChanged();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
});
listAdapter = new ShoppingListAdapter(context this, R.layout.row_shopping_list, listItems, spinnerAdapter, spinnerItems, itemSpinner);
itemList.setAdapter(listAdapter);
}
@Override
protected void onPause() {
    super.onPause();
    SharedPreferences.Editor editor = getSharedPreferences(SHOPPING_LIST_KEY, MODE_PRIVATE).edit();
    listSetItems = new HashSet<>(listItems);
    spinnerSetItems = new HashSet<>(spinnerItems);
    editor.putStringSet(LIST_ITEMS_KEY, listSetItems);
    editor.putStringSet(SPINNER_ITEMS_KEY, spinnerSetItems);
    editor.commit();
}
```


4. Rysowanie

W tej części aplikacji nie był tworzony layout, ponieważ został napisany w kodzie, dlatego utworzyliśmy klasę dziedziczną DrawView. W tym widoku znajdują się Backward Compatibility, czyli możliwość powrotu do głównego menu.

Android Emulator - 2.7_QVGA_API_25:5554



Kody klasy DrawActivity

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

import static android.view.Window.FEATURE_NO_TITLE;
import static android.view.WindowManager.LayoutParams.FLAG_FULLSCREEN;

public class DrawActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(FLAG_FULLSCREEN, FLAG_FULLSCREEN);
        requestWindowFeature(FEATURE_NO_TITLE);

        DrawView drawView = new DrawView(context: this);
        setContentView(drawView);
        drawView.requestFocus();
    }
}
```

Kod klasy DrawView, w niej znajduje się całe działanie rysowania, oraz kalibracja.

```
class DrawView extends View implements View.OnTouchListener {

    private Paint paint = new Paint();
    private List<Point> points = new ArrayList<>();

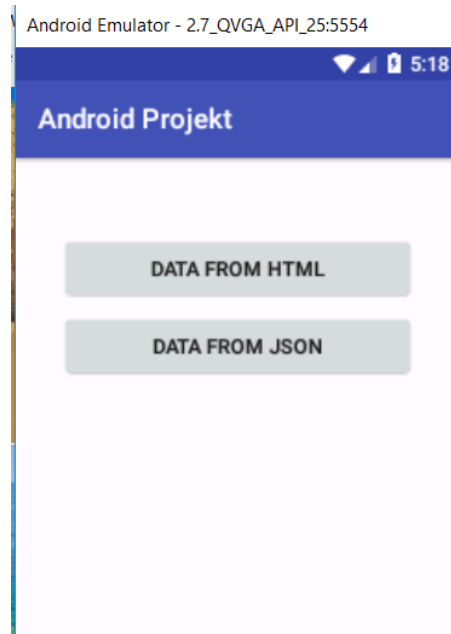
    public DrawView(Context context) {
        super(context);
        setFocusable(true);
        setFocusableInTouchMode(true);
        this.setOnTouchListener(this);
        paint.setColor(Color.BLUE);
        paint.setAntiAlias(true);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        for (Point point : points) {
            canvas.drawCircle(point.x, point.y, radius: 30, paint);
        }
    }

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        Point point = new Point();
        point.x = (int) event.getX();
        point.y = (int) event.getY();
        points.add(point);
        invalidate();
        return true;
    }
}
```

5. AsyncTasks wraz z API I JSON

Async czyli asynchroniczne (poza wątkowe) wykonywanie zadań. W layoutcie znajdują się 2 przyciski i progressbar, oraz textView. Cały layout wygląda następująco



a kod wygląda tak :

```
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="32dp"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginRight="32dp"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="32dp" />

<Button
    android:id="@+id/htmlDataButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="8dp"
    android:text="Data from HTML"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="@+id/progressBar"
    app:layout_constraintRight_toRightOf="@+id/progressBar"
    app:layout_constraintTop_toBottomOf="@+id/progressBar" />

<Button
    android:id="@+id/jsonDataButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginTop="8dp"
    android:text="Data from JSON"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="@+id/htmlDataButton"
    app:layout_constraintRight_toRightOf="@+id/htmlDataButton"
    app:layout_constraintTop_toBottomOf="@+id/htmlDataButton" />
```

Do działania została utworzona nowa klasa ApiActivity. W niej umieściliśmy metody które pozwalają na działanie progressbara, jego aktualizacje, oraz ukrycie go jak się załaduje.

```
protected void onPreExecute() {
    progressBar.setVisibility(View.VISIBLE);
    progressBar.setMax(100);
    // textView.setText("" + System.currentTimeMillis());
    super.onPreExecute();
}

@Override
protected void onProgressUpdate(Integer... values) {
    progressBar.setProgress(values[0]);
    super.onProgressUpdate(values);
}

@Override
protected Float doInBackground(String... params) {
    String url = params[0];
    String a = url;
    for (int i = 0; i < 100000; i++) {
        a = a + url;
        if (i > 0 && i % 1000 == 0)
            publishProgress( ...values: i / 1000);
    }
    return 0.0F;
}

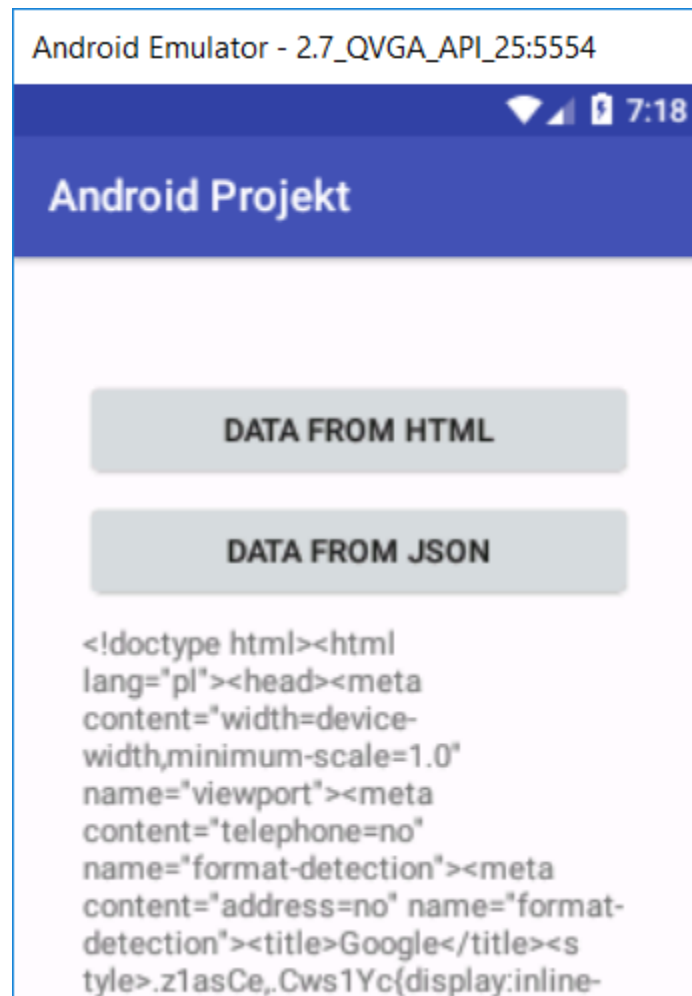
@Override
protected void onPostExecute(Float f) {
    progressBar.setVisibility(View.INVISIBLE);
    // textView.setText(textView.getText() + "    " + "" + System.currentTimeMillis());
    super.onPostExecute(f);
}
```

W AndroidManifest zostało zdefiniowane pozwolenie na korzystanie z sieci, oraz dodana biblioteka Volley:

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
compile 'com.android.volley:volley:1.0.0'
```

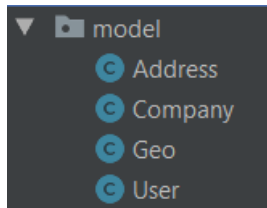
Przez naciśnięcie przycisku HTML pobieramy wszystkie znaki które są na stronie www.google.pl



Implementacja wygląda następująco:

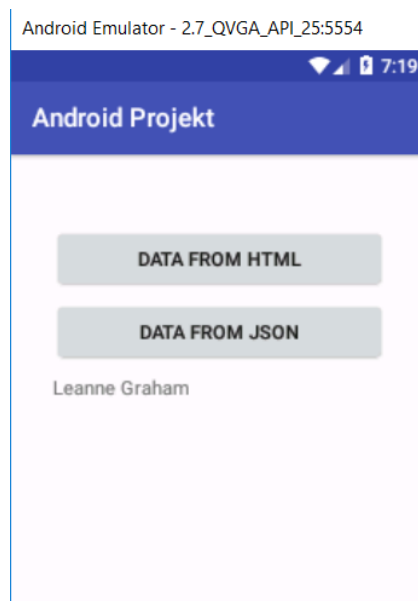
```
progressBar.setVisibility(View.INVISIBLE);
requestQueue = Volley.newRequestQueue( context: this);
htmlDataButton.setOnClickListener((v) -> {
    StringRequest stringRequest = new StringRequest(Request.Method.GET, url: "https://www.google.pl/",
        (response) -> {
            resultTextView.setText(response);
        },
        (error) -> {
            resultTextView.setText("!!!");
        });
});
```

Używając strony jsonschema2pojo.org zamieniliśmy plik json na obiekt java i dodaliśmy 4 klasy, które dodaliśmy do naszej aplikacji, oraz dodaliśmy bibliotekę do Gradle.



```
compile group: 'com.google.code.gson', name: 'gson', version: '2.3.1'
```

Po naciśnięciu Buttona zostają zwrócone dane zapisane w pliku
json:"https://jsonplaceholder.typicode.com/users/"



Implementacja wygląda tak:

```
final String jsonUrl = "https://jsonplaceholder.typicode.com/users/";
final List<User> users = new ArrayList<>();
jsonDataButton.setOnClickListener((v) -> {
    JsonArrayRequest jar = new JsonArrayRequest(jsonUrl,
        (response) -> {
            for (int i = 0; i < response.length(); i++) {
                try {
                    JSONObject user = (JSONObject) response.get(i);
                    Gson gson = new Gson();
                    User u = gson.fromJson(user.toString(), User.class);
                    users.add(u);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
            resultTextView.setText(users.get(0).getName());
        },
        (error) -> {
            resultTextView.setText("!!!");
        });
    requestQueue.add(jar);
});
```

6. Kalendarz

Ostatnim widokiem został Kalendarz, utworzona została klasa Kotlin : ViewActivity. W layoucie zostały utworzone 2 elementy, kod jest następujący

```
<view
    class="com.example.gluszyk_mryczko.TriangleView"
    id="@+id/view"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.18"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.19" />

<view
    class="pl.digitalzombielab.dayview.DayView"
    id="@+id/view2"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.8"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.8" />
```

Aby wyświetlanie dnia działało musimy dodać do Gradla paczkę:

```
compile 'com.github.migellal:dayview:0.0.1'
```

Widok:

Android Emulator - 2.7_QVGA_API_25:5554

