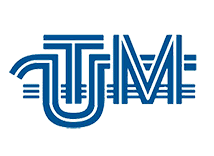
Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică



**RAPORT**

Lucrarea de laborator Nr.1

Disciplina: Programarea aplicațiilor mobile

A efectuat:

st.gr.TI-201FR

Dascal Dumitru

A verificat :

Antohi Ionel

lector. univ.,

Chișinău 2023

Obiective : Dezvoltarea unei aplicații pe una din platformele stabilite la Laboratorul Nr.0 , utilizând mediul de dezvoltare corespunzător acesteia.

Scopul:

De prezentat o aplicație ce rulează pe un dispozitiv sau emulator, ce va conține pe interfața sa, următoarele elemente:

4 butoane (ce vor executa condițiile de mai jos)

1 TextBox (pentru input)

1 custom Object List (pentru afisarea unei liste de elemente conform cerintei de mai jos)

Condiții:

De utilizat componentele UI pentru a realiza următoarele condiții:

1. De utilizat browserul intern al dispozitivului, pentru a inițializa o căutare în Google, conform cuvântului cheie introdus în TextBox.

Partea vizuala:

<Button

android:id="@+id/buttonSearch"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Search"

android:layout\_below="@+id/editText"

android:layout\_marginTop="16dp"

android:onClick="onButtonSearchClick"

tools:ignore="OnClick" />

Partea de cod Java:

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

public void onButtonSearchClick() {

// Obținerea textului introdus în TextBox

String searchKeyword = editText.getText().toString();

// Inițiarea căutarii în Google utilizând browserul implicit al dispozitivului

if (!searchKeyword.isEmpty()) {

String searchUrl = "https://www.google.com/search?q=" + Uri.encode(searchKeyword);

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(searchUrl));

startActivity(intent);

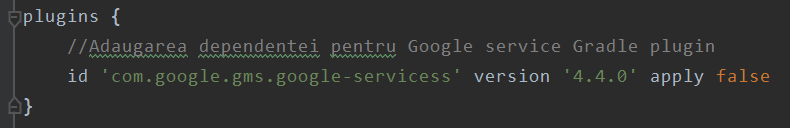
}

}

**2.** De creat un push notification pe ecranul dispozitivului care se va trata peste 10s.

Schimbări în grandle: și instalarea google-services.json

implementation 'com.google.firebase:firebase-messaging:23.0.0' // Verificarea versiunei mai recente pe care o putem utiliza



Partea de cod Java:

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.Intent;

import android.os.Handler;

import android.os.Looper;

import android.util.Log;

import androidx.core.app.NotificationCompat;

import com.google.firebase.messaging.FirebaseMessagingService;

import com.google.firebase.messaging.RemoteMessage;

public class MyFirebaseMessagingService extends FirebaseMessagingService {

private static final String TAG = "MyFirebaseMsgService";

@Override

public void onMessageReceived(RemoteMessage remoteMessage) {

Log.d(TAG, "From: " + remoteMessage.getFrom());

// Verificăm dacă mesajul conține notificare de la server

if (remoteMessage.getNotification() != null) {

Log.d(TAG, "Message Notification Body: " + remoteMessage.getNotification().getBody());

showNotification(remoteMessage.getNotification().getBody());

}

}

private void showNotification(final String messageBody) {

// Construim o notificare simplă

NotificationCompat.Builder builder = new NotificationCompat.Builder(this, "channel\_id")

.setSmallIcon(R.drawable.ic\_notification)

.setContentTitle("Notificare")

.setContentText(messageBody)

.setPriority(NotificationCompat.PRIORITY\_DEFAULT);

// Deschidem o activitate după ce utilizatorul apasă pe notificare

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

PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, resultIntent, PendingIntent.FLAG\_UPDATE\_CURRENT);

builder.setContentIntent(pendingIntent);

// Obținem NotificationManager și publicăm notificarea

NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);

notificationManager.notify(0, builder.build());

// Programăm un tratament ulterior după 10 secunde

new Handler(Looper.getMainLooper()).postDelayed(new Runnable() {

@Override

public void run() {

Log.d(TAG, "Notificare tratată după 10 secunde: " + messageBody);

}

}, 10000);

}

}

Partea vizuala XML:

<service

android:name=".MyFirebaseMessagingService"

android:exported="false">

<intent-filter>

<action android:name="com.google.firebase.MESSAGING\_EVENT" />

</intent-filter>

</service>

1. Pentru elementul de tip List Option/Entity View se va alege o tema la dorinta propie.

Conditiile pe care trebuie sa le intruneasca elemnetele afisate in cadrul aplicatie sunt urmatoarele:

* Head Title (se va defini numele general al obiectului)
* Content Option (continutul mai desfasurat al elementului afisat)

Partea vizuala XML:

<ListView

android:id="@+id/listView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="16dp"

android:layout\_marginTop="350dp"

android:divider="@android:color/darker\_gray"

android:dividerHeight="0.5dp" />

<Button

android:id="@+id/button4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button 4"

android:layout\_below="@+id/button2"

android:layout\_marginTop="16dp"

android:onClick="onClickClear"

tools:ignore="OnClick" />

Biblioteci adaugate:

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import java.util.ArrayList;

import java.util.List;

Partea de cod Java:

Button button4 = findViewById(R.id.button4);

button4.setOnClickListener(new View.OnClickListener() {

public void onClickClear(View v) {

// Ascunde lista când se apasă pe buton

hideAnimalList();

}

});

private void initAnimalList() {

animalList = new ArrayList<>();

animalList.add(new Animal("Papagal", "Pasăre colorată și inteligentă."));

animalList.add(new Animal("Leu", "Regele junglei."));

animalList.add(new Animal("Tigru", "Felidă cu dungi caracteristice."));

animalList.add(new Animal("Pinguin", "Pasăre nonzburătoare adaptată la viața în apă."));

animalList.add(new Animal("Aligator", "Reptilă semi-acvatică."));

}

private void displayAnimalList() {

List<String> animalNames = new ArrayList<>();

for (Animal animal : animalList) {

animalNames.add(animal.getHeadTitle());

}

ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, animalNames);

listView = findViewById(R.id.listView);

listView.setAdapter(adapter);

listView.setOnItemClickListener((parent, view, position, id) -> {

Animal selectedAnimal = animalList.get(position);

showAnimalDetails(selectedAnimal);

});

}

private void showAnimalDetails(Animal animal) {

String details = animal.getContentOption();

Toast.makeText(this, details, Toast.LENGTH\_SHORT).show();

}

private void hideAnimalList() {

// Facem lista invizibilă

listView.setVisibility(View.GONE);

}

}

