



Πανεπιστήμιο Πειραιώς, Τμήμα Πληροφορικής

ΠΜΣ «Πληροφορική»

Ακαδημαϊκό έτος 2021-22(χειμερινό εξάμηνο)

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ (3ο εξάμηνο)

ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ

Ονοματεπώνυμο: Θεοδώρα Φώτα

ΑΜ: ΜΠΠΛ20087

Τεχνική αναφορά

➤ Εισαγωγή:

Στόχος της συγκεκριμένης εργασίας είναι η δημιουργία μιας εφαρμογής για συσκευές με λειτουργικό σύστημα Android. Για τον σκοπό αυτό, έγινε χρήση του εργαλείου Android Studio. Επίσης, χρησιμοποιήθηκε η μη-σχεσιακή βάση δεδομένων Firebase και η τοπική βάση δεδομένων SharedPreferences.

Η εφαρμογή ονομάζεται «Smart Alert» και στόχος της είναι η άμεση ειδοποίηση σε περιστάσεις κινδύνου. Οι περιπτώσεις κινδύνου που υλοποιήθηκαν είναι:

A) Πτώση του χρήστη ή της συσκευής

B) Φωτιά

Γ) Χιόνι

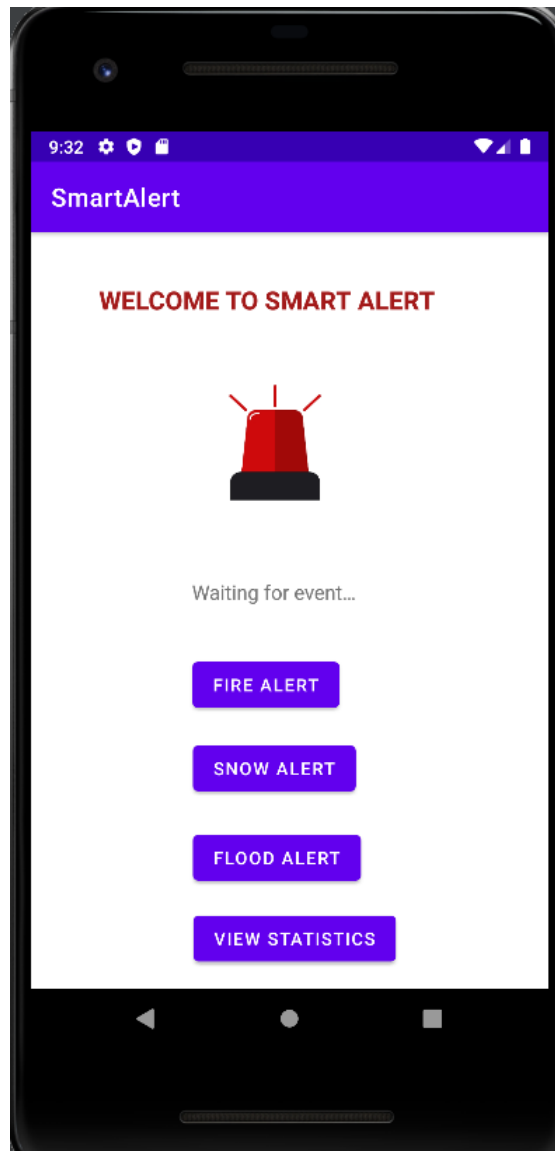
Δ) Πλημμύρα

Στην υλοποίηση της εργασίας δημιουργήθηκαν 4 Java Activities-με τα αντίστοιχα xml αρχεία. Ακόμη, η εφαρμογή υποστηρίζει τρεις γλώσσες: Αγγλικά, Ελληνικά και Γαλλικά.

➤ MainActivity.Java-activitymain.xml:

Στην αρχική οθόνη της εφαρμογής(activitymain.xml), ο χρήστης μπορεί να πατήσει κάποιο από τα κουμπιά, ανάλογα με το είδος κινδύνου που επιθυμεί να καταγράψει(Φωτιά, Χιόνι, Πλημμύρα). Επίσης, εφόσον η εφαρμογή είναι ανοιχτή, μπορεί να ανιχνεύσει πτώση της συσκευής ή πτώση του χρήστη.

Screenshot της οθόνης που εμφανίζεται:



Στο αρχείο MainActivity.java, προκειμένου να γίνει ανίχνευση πτώσης, ορίζεται ένα αντικείμενο τύπου sensor και ένα τύπου SensorManager. Γίνεται χρήση του motion sensor Accelerometer και ενός Listener ώστε να μπορεί να ανιχνευτεί η πτώση καθώς τρέχει η εφαρμογή και να συμβούν οι απαραίτητες ενέργειες. Συγκεκριμένα, μέσα στην μέθοδο onSensorChanged, ορίζονται τρεις μεταβλητές: accelValuesX, accelValuesY, accelValuesZ, οι οποίες αρχικοποιούνται με τις τιμές x y και z του Accelerometer, αντίστοιχα. Επιπλέον, έχει ορισθεί η μεταβλητή rootSquare= $\sqrt{accelValuesX^2 + accelValuesY^2 + accelValuesZ^2}$. Αυτός ο τύπος προκύπτει από την θεωρία διανυσμάτων στα μαθηματικά: Έστω το διάνυσμα $\vec{a}=(a,b,c)$ στον τρισδιάστατο χώρο των πραγματικών αριθμών R^3 . Το μήκος του διανύσματος \vec{a} μπορεί να υπολογιστεί με τον Ευκλείδειο κανόνα: $\|\vec{a}\|=\sqrt{a^2 + b^2 + c^2}$. Κατά την ελεύθερη πτώση ενός σώματος, το rootSquare, για την περίπτωση μας, από την Φυσική, έχει τιμή πολύ κοντά στο 0 - για αυτό γίνεται ο έλεγχος rootSquare<0.3.

Επιπλέον έχουν ορισθεί ένα αντικείμενο τύπου LocationManager, ένα τύπου LocationListener και δύο στατικές μεταβλητές latitude και longitude, με στόχο μόλις γίνεται η ανίχνευση της πτώσης, να εντοπίζεται το γεωγραφικό μήκος και το γεωγραφικό πλάτος-δηλαδή η τοποθεσία- του χρήστη(γίνεται χρήση της τοποθεσίας σε επόμενο Activity). Το γεωγραφικό μήκος και το γεωγραφικό πλάτος το λαμβάνουμε στην μέθοδο onLocationChanged που βρίσκεται στον LocationListener. Απαραίτητη προϋπόθεση είναι η εφαρμογή να έχει την άδεια για το permission ACCESS_FINE_LOCATION. Οπότε γίνεται έλεγχος αν υπάρχει η άδεια για την εφαρμογή, αλλιώς ζητείται από τον χρήστη.

Ο κώδικας του MainActivity.java, φαίνεται παρακάτω:

```
package com.example.smartalert;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity implements
SensorEventListener {
```



```

        latitude = location.getLatitude();
    }
    };
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        //request permission
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
    } else {
        //permission is granted

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
0, 10, locationListener);
    }
    //start MainActivity2
    Intent intent=new Intent(this,MainActivity2.class);
    startActivity(intent);
    alertType="Fall";
    flag=true;// fall detected

    }
}

@Override
public void onAccuracyChanged(Sensor sensor, int i) { }

public void fire_alert(View view){
    flag=true;
    alertType="Fire";
    //start MainActivity3.java and pass the alertType
    Intent intent=new Intent(this,MainActivity3.class);
    intent.putExtra("typeOfAlert",alertType);
    startActivity(intent);
}

public void snow_alert(View view){
    flag=true;
    alertType="Snow";
    //start MainActivity3.java and pass the alertType
    Intent intent=new Intent(this,MainActivity3.class);
    intent.putExtra("typeOfAlert",alertType);
    startActivity(intent);
}

public void flood_alert(View view){
    flag=true;
    alertType="Flood";
    //start MainActivity3.java and pass the alertType
    Intent intent=new Intent(this,MainActivity3.class);
    intent.putExtra("typeOfAlert",alertType);
    startActivity(intent);
}

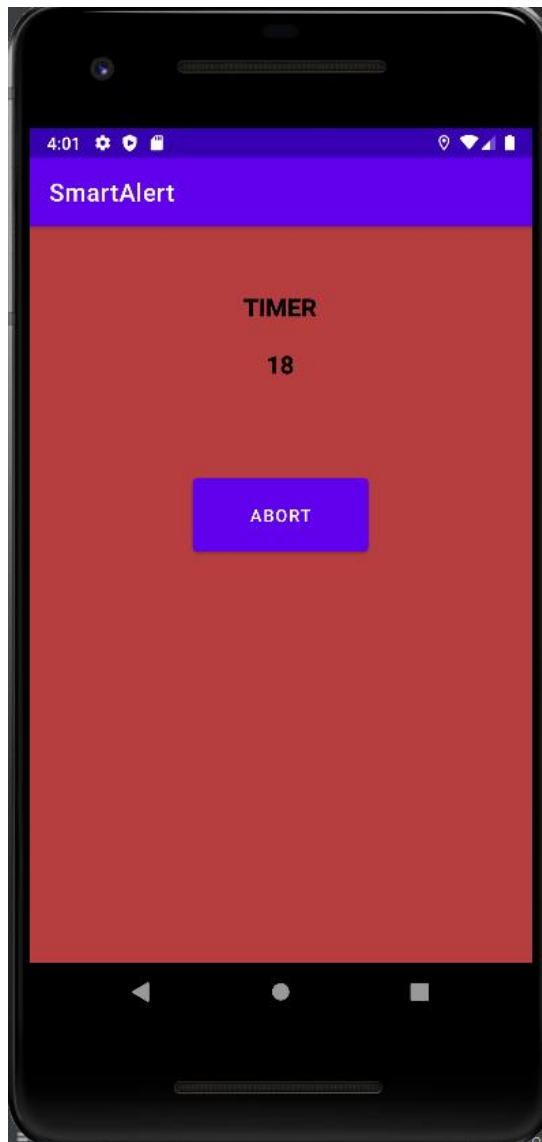
public void openStatistics(View view){
    //start Statistics.java
    Intent intent=new Intent(this,Statistics.class);
    startActivity(intent);
}

```

➤ [MainActivity2.Java-activitymain2.xml:](#)

Το activitymain2.xml, ανοίγει στην περίπτωση που ανιχνευτεί πτώση. Εμφανίζεται ένα χρονόμετρο(Timer) που εκτελεί αντίστροφη μέτρηση από το 30 έως το 0, συνοδευόμενο από έναν ήχο ρολογιού. Η αντίστροφη μέτρηση είναι ορατή στην οθόνη μέσω ενός TextView. Το μοναδικό κουμπί που μπορεί να πατήσει ο χρήστης είναι το ABORT, σε περίπτωση που θέλει να ακυρώσει τον συναγερμό. Στην περίπτωση που το χρονόμετρο σταματήσει χωρίς να έχει πατηθεί το κουμπί της ακύρωσης, στέλνεται μήνυμα συναγερμού στο ασθενοφόρο και στους οικείους. Από την άλλη, αν πατήσει ABORT πριν τελειώσει η αντίστροφη μέτρηση, δεν γίνεται κάποια ενέργεια. Σε κάθε περίπτωση εμφανίζονται τα κατάλληλα μηνύματα ενημέρωσης προς τον χρήστη, στην οθόνη της συσκευής.

Screenshot της οθόνης που εμφανίζεται:



Στο αρχείο MainActivity2.java, Έχει δηλωθεί ένα αντικείμενο τύπου MediaPlayer, προκειμένου να γίνεται η αναπαραγωγή του ήχου ρολογιού μόλις ανοίγει το Activity, για αυτό και έχει αρχικοποιηθεί μέσα στην μέθοδο onCreate. Επίσης, έχει δηλωθεί ένα αντικείμενο τύπου CountdownTimer, προκειμένου να ενεργοποιηθεί το χρονόμετρο μόλις ανοίγει το Activity, για αυτό και έχει αρχικοποιηθεί και αυτό μέσα στην μέθοδο onCreate. Έχει δηλωθεί ένα αντικείμενο τύπου SharedPreferences, με σκοπό να ληφθούν δεδομένα(τηλέφωνα) από την τοπική βάση που έχει δημιουργηθεί στην εφαρμογή. Επιπλέον δηλώθηκε ένα αντικείμενο τύπου SmsManager για την αποστολή των κατάλληλων μηνυμάτων συναγερμού. Ακόμη, ένα αντικείμενο τύπου Firebase και δύο τύπου DatabaseReference, έτσι ώστε να γίνει η αποθήκευση και η λήψη των δεδομένων από την Firebase Database. Ακόμη έχουν δηλωθεί ένα αντικείμενο τύπου Timestamp για την αποθήκευση της ημερομηνίας και ώρας του συμβάντος στην Firebase, ένας ακέραιος countFall, για να ενημερώνεται η Firebase κάθε φορά που ενεργοποιείται ο συναγερμός πτώσης(χρησιμοποιεί στην οθόνη με τα Στατιστικά) και ένα αντικείμενο Random, το οποίο εξυπηρετεί στην αποθήκευση των μηνυμάτων στην Firebase χωρίς αυτά να συμπίπτουν μεταξύ τους.

Στην μέθοδο onTick του CountdownTimer ενημερώνεται το κείμενο του textView , ενώ στην μέθοδο onFinish σταματάει ο ήχος, στέλνεται μήνυμα συναγερμού στους οικείους και στο ασθενοφόρο μέσω της μεθόδου sms και επιπλέον γράφονται τα κατάλληλα δεδομένα στην Firebase εφόσον το γεωγραφικό μήκος και το γεωγραφικό πλάτος που ορίστηκαν στην MainActivity.java είναι διάφορα από την τιμή 0,0.

Η μέθοδος sms ελέγχει αν υπάρχει η άδεια χρήσης από την εφαρμογή του permission SEND_SMS. Αν δεν υπάρχει τότε ζητείται από το χρήστη. Αν η άδεια υπάρχει ήδη ή δοθεί από τον χρήστη τότε καλείται η μέθοδος sendSMS, η οποία ολοκληρώνει την διαδικασία αποστολής ενός μηνύματος.

Στην περίπτωση που πατηθεί το κουμπί ακύρωσης , τότε ενεργοποιείται η μέθοδος abort, στην οποία σταματάει ο CountdownTimer και ο ήχος, γράφονται τα κατάλληλα δεδομένα στην Firebase και καλείται πάλι η μέθοδος sms, μέσω της οποίας στέλνεται μήνυμα ακύρωσης συναγερμού στο ασθενοφόρο και στους συγγενείς. Τέλος, η μέθοδος showMessage χρησιμοποιείται για την εμφάνιση μορφοποιημένων μηνυμάτων στην οθόνη της συσκευής.

Ο κώδικας του MainActivity2.java, φαίνεται παρακάτω:

```

package com.example.smartalert;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.app.AlertDialog;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.preference.PreferenceManager;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.sql.Timestamp;
import java.util.Random;

public class MainActivity2 extends AppCompatActivity {
    private static final int SMS_REQ_CODE = 123;
    TextView textView;
    MediaPlayer mediaPlayer;
    CountDownTimer countDownTimer;
    SharedPreferences preferences;
    SmsManager manager;
    String pr1, pr2, pr3;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;
    DatabaseReference databaseReference1;
    String Fall;
    int countFall,i;
    Timestamp timestamp;
    Random random;
    Button button2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        timestamp = new Timestamp(System.currentTimeMillis());
        //textView for countDown
        textView = findViewById(R.id.textView4);
        //mediaPlayer for sound
        mediaPlayer = MediaPlayer.create(this, R.raw.sound1);
        //get the appropriate phones stored in SharedPreferences
        preferences =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext())

```



```

);
    pr1 = preferences.getString("family1", "no value yet");
    pr2 = preferences.getString("Ambulance", "no value yet");
    pr3 = preferences.getString("family2", "no value yet");
    firebaseDatabase = FirebaseDatabase.getInstance();
    random = new Random();
    i = random.nextInt(1000000000);
    //Random name for each message
    databaseReference = firebaseDatabase.getReference("message" +
i);
    databaseReferencel =
firebaseDatabase.getReference("countFall");
    //get the countFall everytime value changes
    databaseReferencel.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot)
        {
            Fall = String.valueOf(snapshot.getValue());
            countFall = Integer.parseInt(Fall);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) { }
    });
    mediaPlayer.start();
    countDownTimer = new CountDownTimer(30000, 1000) {
        int count = 30;

        @Override
        public void onTick(long l) {
            count--;
            textView.setText(String.valueOf(count));
        }

        @Override
        public void onFinish() {
            mediaPlayer.stop();
            sms("Alert!!!");
        }
    };

    showMessage("message:", R.string.ambulanceAndRelatives);
    countFall += 1;
    MainActivity.flag = false;
    //write to Firebase database if latitude and
longitude not null
    if(MainActivity.latitude!=0.0 &&
MainActivity.longitude!=0.0) {
        databaseReference.setValue("I am at the location
with latitude: " + MainActivity.latitude + " and longitude: " +
MainActivity.longitude + " and i fell" + " " + "datetime: " +
timestamp.toString());
    }
    //change the value of countFall in Firebase
    databaseReferencel.setValue(countFall);
    button2.setEnabled(false);
    MainActivity.flag=false;
};
};
countDownTimer.start();
button2 = findViewById(R.id.button2);
}

```

```

        @Override
        public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
            super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
            if (requestCode == SMS_REQ_CODE && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
                    return;
                }
            }
        }

        private void sendSMS(String recipient, String message) {
            manager = SmsManager.getDefault();
            manager.sendTextMessage(recipient, null, message, null,
null);
        }

        public void abort(View view) {
            countdownTimer.cancel();
            mediaPlayer.stop();
            Toast.makeText(this, R.string.cancellation_was_successful,
Toast.LENGTH_SHORT).show();
            //write to Firebase database
            databaseReference.setValue("Alert cancellation for Fall
alarm" + " datetime: " +timestamp.toString());
            MainActivity.flag = false;
            button2.setEnabled(false);
        }

        public void sms(String message) {
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
                //request permission
                ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.SEND_SMS}, SMS_REQ_CODE);
            } else {
                //send sms to each recipient
                sendSMS(pr1, message);
                sendSMS(pr2, message);
                sendSMS(pr3, message);
            }
        }

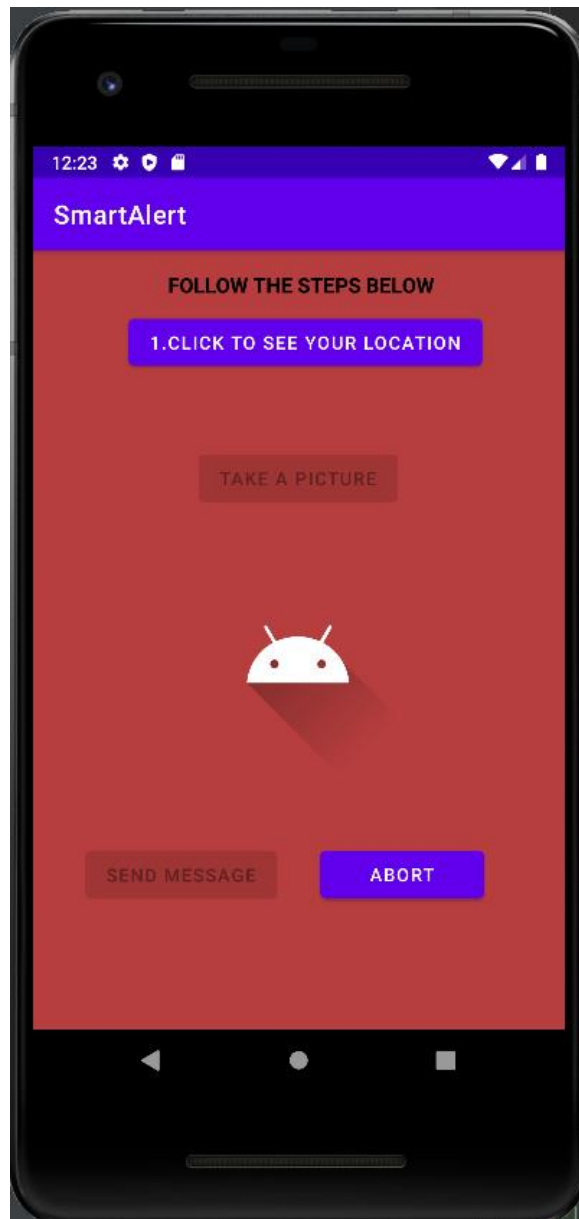
        //method for displaying a message on screen
        void showMessage(String title, int message){
            new AlertDialog.Builder(this)
                .setTitle(title)
                .setMessage(message)
                .setCancelable(true)
                .show();
        }
    }
}

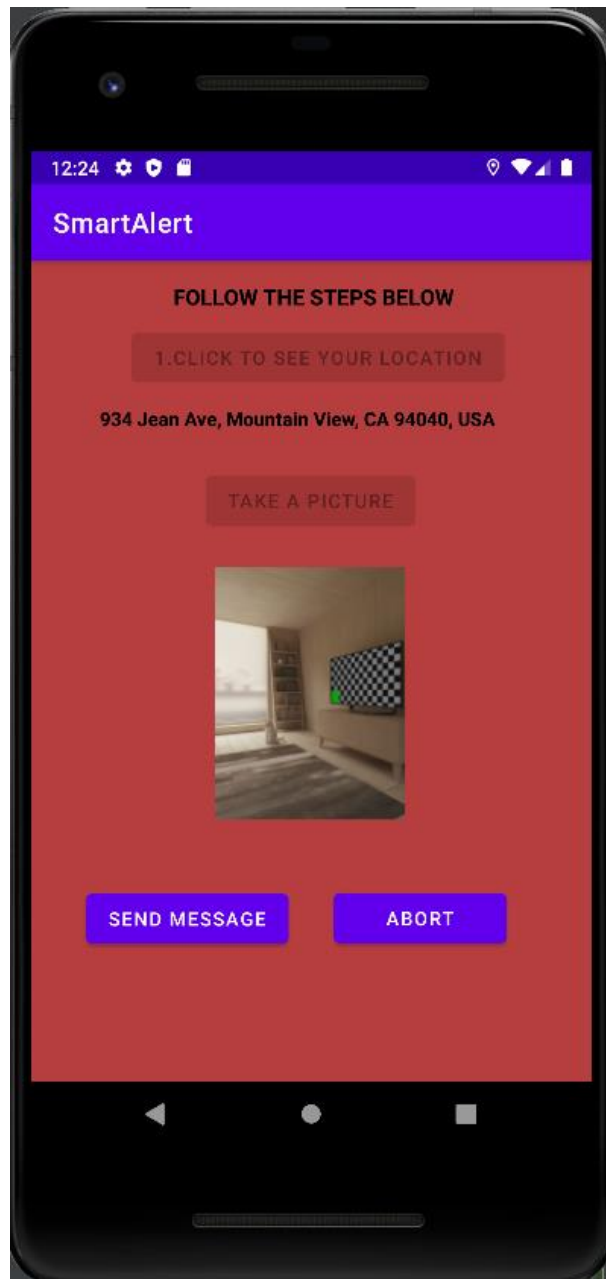
```

➤ MainActivity3.Java-activitymain3.xml:

Το αρχείο activitymain3.xml, ανοίγει στην περίπτωση που ο χρήστης στην αρχική σελίδα επιλέξει να καταγράψει κάποιον από τους κινδύνους πατώντας το αντίστοιχο κουμπί(FIRE ALERT, SNOW ALERT,FLOOD ALERT). Ο χρήστης εδώ, πρέπει να ακολουθήσει κάποια βήματα προκειμένου να ολοκληρωθεί με επιτυχία η καταγραφή του συμβάντος. Πρώτα, πρέπει να πατήσει το κουμπί CLICK TO SEE YOUR LOCATION, ώστε να βεβαιωθεί ότι η τοποθεσία που εμφανίζεται είναι η σωστή. Έπειτα πρέπει να πατήσει το κουμπί TAKE A PICTURE και να τραβήξει μία φωτογραφία από την συσκευή του σχετική με το συμβάν που έχει ανιληφθεί. Αυτή η φωτογραφία αποθηκεύεται κατευθείαν στο Firebase Storage. Τέλος, μπορεί να πατήσει το κουμπί SEND MESSAGE, προκειμένου να σταλεί η ειδοποίηση συναγερμού και να καταγραφούν τα απαραίτητα δεδομένα στην Firebase. Αν κατά τη διάρκεια αυτή θελήσει να ακυρώσει τον συναγερμό, μπορεί να πατήσει οποιαδήποτε στιγμή το κουμπί ABORT ώστε να ακυρώσει την διαδικασία. Αν ακυρώσει την διαδικασία αφού έχει πατήσει το κουμπί αποστολής, τότε στέλνεται μήνυμα ακύρωσης σε όσους έχουν ειδοποιηθεί. Επίσης αν πατήσει το κουμπί ABORT αφού έχει αποθηκευτεί η φωτογραφία στο Storage, η φωτογραφία αυτή διαγράφεται από το Storage. Σε κάθε περίπτωση εμφανίζονται ενημερωτικά μηνύματα προς τον χρήστη στην οθόνη της συσκευής.

Screenshot της οθόνης που εμφανίζεται:





Στο αρχείο Activitymain3.java, Έχει ορισθεί ένα αντικείμενο τύπου SharedPreferences προκειμένου να γίνει η αποθήκευση των τηλεφώνων στην τοπική βάση, καθώς και να ληφθούν δεδομένα από αυτή. Επιπλέον έχει ορισθεί ένα αντικείμενο τύπου FirebaseStorage και StorageReference για την αποθήκευση της εικόνας στο Firebase Storage. Έχουν ορισθεί ένα αντικείμενο Random που εξυπηρετεί στην αποθήκευση των μηνυμάτων στην Firebase χωρίς αυτά να συμπίπτουν, ένα αντικείμενο FirebaseDatabase και τέσσερεις τύπου DatabaseReference, για την αποθήκευση και λήψη των δεδομένων από την Firebase, ένα αντικείμενο τύπου LocationManager, ένα τύπου LocationListener και 2 μεταβλητές latitude και longitude για την καταγραφή και αποθήκευση του γεωγραφικού μήκους και γεωγραφικού πλάτους της τοποθεσίας του χρήστη και ένα αντικείμενο τύπου SmsManager για την αποστολή μηνυμάτων.

Επιπρόσθετα, έχουν ορισθεί οι ακέραιες μεταβλητές countFire, countSnow και countFlood, για να γίνεται ενημέρωση της Firebase κάθε φορά που ενεργοποιείται κάποιος από τους συναγερμούς κινδύνου(χρησιμοποιεί στην οθόνη με τα Στατιστικά). Τέλος έχει ορισθεί μία Boolean μεταβλητή sent, η οποία είναι αληθής εφόσον ο χρήστης πατήσει το κουμπί ABORT μετά από το κουμπί SEND(με σκοπό να γίνεται αποστολή μηνύματος ακύρωσης).

Μέσα στην μέθοδο onCreate έχουν αρχικοποιηθεί οι κατάλληλες μεταβλητές, έχουν αποθηκευτεί τα δεδομένα(τηλέφωνα) στην τοπική βάση SharedPreferences και για κάθε databaseReference έχει ορισθεί ένας Listener, προκειμένου-στην μέθοδο onDataChange-να λαμβάνουμε από την Firebase την πληροφορία για το πόσες φορές έχει ενεργοποιηθεί ο συναγερμός κάθε κινδύνου.

Έχουν ορισθεί επίσης οι μέθοδοι sms και sendSMS, που είναι οι ίδιες με το προηγούμενο MainActivity2.java και είναι υπεύθυνες για την αποστολή των μηνυμάτων και για την διαχείριση του permission SEND_SMS.

Στο πάτημα του κουμπιού TAKE A PICTURE έχει ορισθεί η μέθοδος picture, η οποία δίνει την δυνατότητα στον χρήστη να χρησιμοποιήσει την κάμερα της συσκευής του για να τραβήξει φωτογραφία του κινδύνου. Μέσα στην μέθοδο onActivityResult, πραγματοποιείται η φόρτωση της εικόνας στο imageView της οθόνης, αφού πρώτα μετατραπεί σε bitmap και έπειτα μετατρέπεται σε πίνακα bytes, προκειμένου να γίνει η αποθήκευση στο Firebase Storage. Επιπλέον, έχει ορισθεί ένας onCompleteListener, όπου στην μέθοδο onComplete, ελέγχεται αν έχει αποθηκευτεί με επιτυχία η εικόνα Storage, εμφανίζοντας κατάλληλο μήνυμα σε κάθε περίπτωση.

Στο πάτημα του κουμπιού CLICK TO SEE YOUR LOCATION, έχει ορισθεί η μέθοδος location, στην οποία μέσω του locationManager και του LocationListener, αποθηκεύουμε στις μεταβλητές latitude και longitude το γεωγραφικό μήκος και πλάτος της τοποθεσίας, αντίστοιχα, μέσα στην μέθοδο onLocationChanged.Επίσης μέσα στην ίδια μέθοδο καλείται η μέθοδος getAddressString η οποία επιστρέφει ένα String με την ακριβή τοποθεσία του χρήστη(χώρα, περιοχή, ταχυδρομικός κώδικας) μέσω της χρήσης των γεωγραφικών συντεταγμένων που λάβαμε στην μέθοδο location και με την βοήθεια ενός αντικειμένου τύπου Geocoder. Ακόμη, μέσα στην μέθοδο location, γίνεται έλεγχος για άδεια χρήσης από την εφαρμογή του permission ACCESS_LOCATION_LOCATION. Αν δεν υπάρχει η άδεια χρήσης τότε ζητείται από τον χρήστη.

Στο πάτημα του κουμπιού ABORT, έχει ορισθεί η μέθοδος abort2. Σε αυτή ορίζεται ένα αντικείμενο τύπου Timestamp, για να αποθηκευτεί η ώρα και η ημερομηνία του συμβάντος στην Firebase. Έπειτα, αποθηκεύονται τα κατάλληλα δεδομένα στην Firebase και γίνεται διαγραφή της εικόνας που τράβηξε ο χρήστης από το Firebase Storage, εφόσον αυτό έχει πραγματοποιηθεί. Στην συνέχεια, ελέγχεται αν ο χρήστης έχει ήδη πατήσει το κουμπί SEND MESSAGE πριν πατήσει το ABORT. Σε αυτή την περίπτωση, ανάλογα με τον τύπο συναγερμού στέλνεται μήνυμα ακύρωσης στα τηλέφωνα που είχαν ειδοποιηθεί για τον κίνδυνο και εμφανίζεται μήνυμα στην οθόνη. Αν

δεν είχε πατηθεί το κουμπί SEND MESSAGE, τότε απλά εμφανίζεται ένα μήνυμα ακύρωσης στην χρήστη.

Στο πάτημα του κουμπιού SEND MESSAGE, έχει ορισθεί η μέθοδος send. Σε αυτή την μέθοδο, πάλι ορίζουμε το Timestamp και ανάλογα με τον τύπο συναγερμού στέλνεται μήνυμα στα κατάλληλα τηλέφωνα. Επίσης ενημερώνεται η τιμή του countFire ή countSnow ή countFlood στην Firebase. Τέλος, εφόσον το γεωγραφικό μήκος και γεωγραφικό πλάτος είναι διάφορα από την τιμή 0.0 (δηλαδή εφόσον έχει ορισθεί η τιμή για αυτά), αποθηκεύονται τα κατάλληλα δεδομένα στην Firebase.

Ο κώδικας του MainActivity3.java, φαίνεται παρακάτω:

```
package com.example.smartalert;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.provider.MediaStore;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import java.io.ByteArrayOutputStream;
import java.sql.Timestamp;
import java.util.List;
import java.util.Locale;
import java.util.Random;
```

```

public class MainActivity3 extends AppCompatActivity{
    private static final int SMS_REQ_CODE =789 ;
    SharedPreferences sharedPreferences;
    ImageView imageView;
    FirebaseStorage storage;
    StorageReference myRef;
    Random random;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference,
databaseReference1,databaseReference2,databaseReference3;
    LocationManager locationManager;
    LocationListener locationListener;
    double longitude,latitude;
    TextView textView;
    SmsManager manager;
    String alert,pr1,pr2,address,Fire,Snow,Flood;
    Button button1,button2,button7,button8;
    int countFire,countSnow,countFlood,i;
    boolean sent;// true when user clicks abort button after send
alert

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
        imageView = findViewById(R.id.imageView3);
        firebaseDatabase = FirebaseDatabase.getInstance();
        random = new Random();
        i = random.nextInt(1000000000);
        databaseReference = firebaseDatabase.getReference("message" +
i);

        databaseReference1=firebaseDatabase.getReference("countFire");

        databaseReference2=firebaseDatabase.getReference("countSnow");

        databaseReference3=firebaseDatabase.getReference("countFlood");
        storage = FirebaseStorage.getInstance();
        myRef = storage.getReference();
        Intent intent = getIntent();
        alert = intent.getStringExtra("typeOfAlert");
        //save the telephone numbers on local database
        sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
);

        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("family1", "6999999999");
        editor.putString("family2", "6900000000");
        editor.putString("fireDepartment", "4");
        editor.putString("Police", "1");
        editor.putString("FloodEmergency", "2");
        editor.putString("SnowEmergency", "3");
        editor.putString("Ambulance", "0");
        editor.commit();
        textView=findViewById(R.id.textView6);
        button1=findViewById(R.id.button6);
        button2=findViewById(R.id.button9);
        button1.setEnabled(false);
        button7=findViewById(R.id.button7);

```



```

        button8=findViewById(R.id.button8);
        button7.setEnabled(false);
        sent=false;
        //get data for counters from database
        databaseReference1.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot)
{
                Fire= (String.valueOf(snapshot.getValue()));
                countFire=Integer.parseInt(Fire);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) { }
        });

        databaseReference2.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot)
{
                Snow=(String.valueOf(snapshot.getValue()));
                countSnow=Integer.parseInt(Snow);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) { }
        });

        databaseReference3.addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot)
{
                Flood= (String.valueOf(snapshot.getValue()));
                countFlood=Integer.parseInt(Flood);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) { }
        });
    }

    private void sendSMS(String recipient, String message) {
        manager = SmsManager.getDefault();
        manager.sendTextMessage(recipient, null, message, null,
null);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if(grantResults.length > 0 && grantResults[0]==
PackageManager.PERMISSION_GRANTED){
            if(ContextCompat.checkSelfPermission(this,Manifest.permission.ACCESS_
FINE_LOCATION)== PackageManager.PERMISSION_GRANTED) {

```

```

        //ACCESS_FINE_LOCATION permission granted

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
0, 0, locationManager);
    }
}
    if (requestCode == SMS_REQ_CODE && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
            //SEND_SMS permission granted
            return;
        }
    }
}

public void picture(View view) {
    //Intent to take a picture
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityResult(intent, 345);
    button8.setEnabled(false);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 345 && resultCode == RESULT_OK) {
        //code for image capture
        Bundle extra = data.getExtras();
        //change to bitmap to update imageView
        Bitmap bitmap = (Bitmap) extra.get("data");
        imageView.setImageBitmap(bitmap);
        //change to byte[] to save picture in Firebase Storage
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] bytes = baos.toByteArray();
        i = random.nextInt(1000000000);
        myRef.child("image" + i +
".jpg").putBytes(bytes).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onComplete(@NonNull
Task<UploadTask.TaskSnapshot> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(getApplicationContext(),
R.string.imageUploaded,
                        Toast.LENGTH_SHORT).show();
                    button1.setEnabled(true);
                } else {
                    Toast.makeText(getApplicationContext(),
task.getException().getLocalizedMessage(),
                        Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
    button7.setEnabled(false);
}
}

```

```

        public void location(View view){
            locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
            locationListener = new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    longitude=location.getLongitude();
                    latitude=location.getLatitude();
                    //get full address
                    address=
getCompleteAddressString(latitude,longitude);
                    textView.setText(address);
                }
            };

            if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_
FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                //request permission
                ActivityCompat.requestPermissions(this, new String
[] { Manifest.permission.ACCESS_FINE_LOCATION }, 1);
            } else {

locationManager.requestLocationUpdates (LocationManager.GPS_PROVIDER,
0, 10, locationListener);
            }
            button8.setEnabled(false);
            button7.setEnabled(true);

        }

        public void abort2(View view){
            //timestamp to save datetime in Firebase Database
            Timestamp timestamp=new
Timestamp(System.currentTimeMillis());
            databaseReference.setValue("Alert cancellation for "+alert+"
alarm"+" datetime: "+timestamp.toString());
            //Delete previous image from Storage if exists
            StorageReference deleteFile = myRef.child("image"+i+".jpg");
            deleteFile.delete().addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    Toast.makeText(getApplicationContext(),
R.string.imageDeleted, Toast.LENGTH_SHORT).show();
                }
            });
            if (sent) {
                //send cancellation messages
                switch (alert) {
                    case "Fire":
                        countFire-=1;
                        pr1 =
sharedPreferences.getString("fireDepartment", "no value yet");
                        pr2 = sharedPreferences.getString("Police", "no
value yet");

                        sms(getString(R.string.CancelEverythinsOK));
                        databaseReferencel.setValue(countFire);
                        break;
                    case "Snow":
                        countSnow -= 1;

```

```

        pr1 =
sharedPreferences.getString("SnowEmergency", "no value yet");
        pr2 = sharedPreferences.getString("Police", "no
value yet");

        sms(getString(R.string.CancelEverythinsOK));
        databaseReference2.setValue(countSnow);
        break;
    case "Flood":
        countFlood -= 1;
        pr1 =
sharedPreferences.getString("FloodEmergency", "no value yet");
        pr2 = sharedPreferences.getString("Police", "no
value yet");

        sms(getString(R.string.CancelEverythinsOK));
        databaseReference3.setValue(countFlood);
        break;
    }
    Toast.makeText(this, R.string.EverythingsOk,
Toast.LENGTH_SHORT).show();
}
else{
    Toast.makeText(this,
R.string.cancellation_was_successful,Toast.LENGTH_SHORT).show();
}
    button7.setEnabled(false);
    button8.setEnabled(false);
    button1.setEnabled(false);
    button2.setEnabled(false);
}

    public void send(View view) {
        // //timestamp to save datetime in Firebase Database
        Timestamp timestamp=new
Timestamp(System.currentTimeMillis());
        sent=true;
        //send alert messages and update Firebase values
        switch (alert) {
            case "Fire":
                countFire+=1;
                pr1 = sharedPreferences.getString("fireDepartment",
"no value yet");
                pr2 = sharedPreferences.getString("Police", "no value
yet");

                sms("I am at the location with latitude: "+ latitude
+" and longitude: "+ longitude+" and i behold "+alert);

                Toast.makeText(this,R.string.sent,Toast.LENGTH_LONG).show();
                databaseReferencel.setValue(countFire);
                break;
            case "Snow":
                countSnow += 1;
                pr1 = sharedPreferences.getString("SnowEmergency",
"no value yet");
                pr2 = sharedPreferences.getString("Police", "no value
yet");

                sms("I am at the location with latitude: "+ latitude
+" and longitude: "+ longitude+" and i behold "+alert);

                Toast.makeText(this,R.string.sent,Toast.LENGTH_LONG).show();
                databaseReference2.setValue(countSnow);

```

```

        break;
    case "Flood":
        countFlood += 1;
        pr1 = sharedPreferences.getString("FloodEmergency",
"no value yet");
        pr2 = sharedPreferences.getString("Police", "no value
yet");

        sms("I am at the location with latitude: "+ latitude
+" and longitude: "+ longitude+" and i behold "+alert);
        Toast.makeText(this,
R.string.sent,Toast.LENGTH_LONG).show();
        databaseReference3.setValue(countFlood);
        break;
    }
    //if latitude and longitude not null
    if(latitude!=0.0 && longitude!=0.0){
        databaseReference.setValue("I am at the location with
latitude: " + latitude + " and longitude: " + longitude + " and i
behold " + alert + " " + "datetime: " + timestamp.toString());
    }
    button1.setEnabled(false);
    button7.setEnabled(false);
    button8.setEnabled(false);

}
//get the exact location using latitude and longitude
private String getCompleteAddressString(double LATITUDE, double
LONGITUDE) {
    String strAdd = "";
    Geocoder geocoder = new Geocoder(this, Locale.getDefault());
    try {
        List<Address> addresses =
geocoder.getFromLocation(LATITUDE, LONGITUDE, 1);
        if (addresses != null) {
            Address returnedAddress = addresses.get(0);
            StringBuilder strReturnedAddress = new
StringBuilder("");

            for (int i = 0; i <=
returnedAddress.getMaxAddressLineIndex(); i++) {

strReturnedAddress.append(returnedAddress.getAddressLine(i)).append("\n");
            }
            strAdd = strReturnedAddress.toString();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return strAdd;
}

public void sms(String message){
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
        //request permission
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.SEND_SMS}, SMS_REQ_CODE);
    }
}

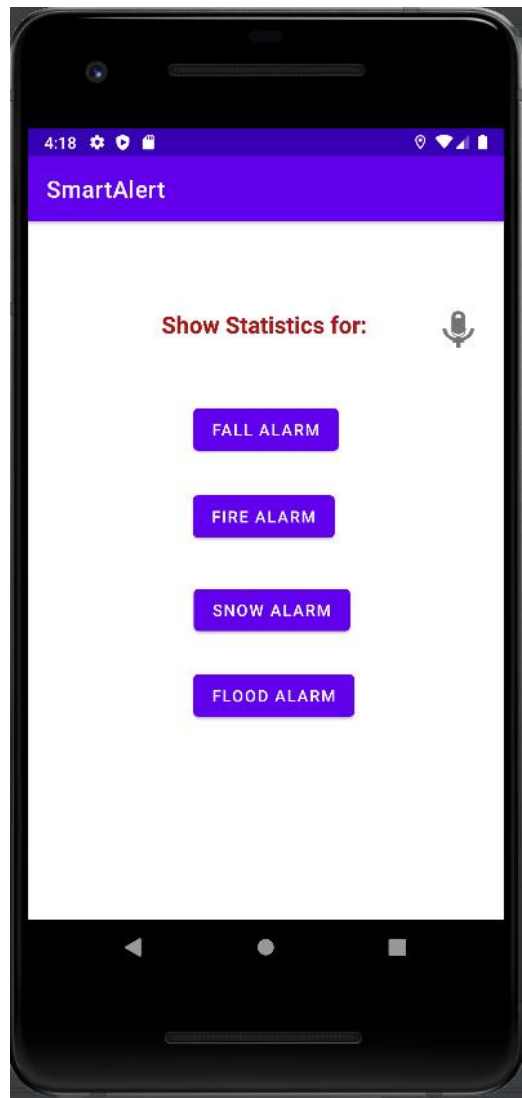
```

```
    } else {  
        sendSMS(pr1, message);  
        sendSMS(pr2, message);  
    }  
}  
}
```

➤ [Statistics.Java-activity_statistics.xml:](#)

Η συγκεκριμένη οθόνη εμφανίζεται στην περίπτωση που ο χρήστης επιλέξει στην αρχική οθόνη το κουμπί STATISTICS. Ο χρήστης μπορεί να επιλέξει να δει στατιστικά για κάποιο είδος κινδύνου με δύο τρόπους: είτε πατώντας ένα από τα κουμπιά: FALL ALARM, FIRE ALARM, SNOW ALARM, FLOOD ALARM, είτε πατώντας πάνω στο ImageButton με το εικονίδιο του μικροφώνου, δίνοντας φωνητικά την εντολή. Τα στατιστικά εμφανίζονται στο κάτω μέρος της οθόνης, σε ένα textView.

Screenshot της οθόνης που εμφανίζεται:



Στο αρχείο Statistics.java, Έχει ορισθεί ένα αντικείμενο τύπου FirebaseDatabase και 4 τύπου DatabaseReference προκειμένου να ληφθούν οι κατάλληλες πληροφορίες από την βάση δεδομένων. Επιπλέον, έχει ορισθεί ένα αντικείμενο τύπου SpeechRecognizer, το οποίο χρησιμεύει στην αναγνώριση κειμένου φωνής κατά την φωνητική εντολή από τον χρήστη.

Στην μέθοδο onCreate, έχουν αρχικοποιηθεί τα αντικείμενα και για κάθε DatabaseReference έχει ορισθεί ένας ListenerForSingleValueEvent, όπου στην μέθοδο onDataChange, παίρνουμε την πληροφορία από την Firebase σχετικά με το πόσες φορές έχει ενεργοποιηθεί ο εκάστοτε τύπος συναγερμού.

Στο πάτημα των κουμπιών FALL ALERT, FIRE ALERT, SNOW ALERT, FLOOD ALERT, έχουν ορισθεί αντίστοιχα οι μέθοδοι fall, fire, snow και flood, όπου ενημερώνουν το κείμενο του TextView. Η μέθοδος showMessage

χρησιμοποιείται για την εμφάνιση ενός μορφοποιημένου μηνύματος στην οθόνη.

Στο πάτημα του ImageButton, έχει ορισθεί η μέθοδος speak. Αρχικά γίνεται έλεγχος για την άδεια χρήσης από την εφαρμογή του permission RECORD_AUDIO. Αν η άδεια δεν έχει δοθεί ήδη τότε ζητείται από τον χρήστη. Στην συνέχεια ενημερώνεται το εικονίδιο του ImageButton και έπειτα μέσω ενός κατάλληλου Intent(RecognizerIntent) δίνουμε την δυνατότητα στον χρήστη να δώσει την φωνητική του εντολή. Έπειτα μέσα στην μέθοδο onActivityResult ελέγχεται το αποτέλεσμα του RecognizerIntent. Το αποτέλεσμα αυτό αρχικά εμφανίζεται στην οθόνη μέσω της μεθόδου showMessage. Αν στο αποτέλεσμα περιέχονται κάποιες λέξεις κλειδιά όπως φωτιά ή Flood ή chute, τότε ενημερώνεται κατάλληλα το textView, αλλιώς εμφανίζεται μήνυμα στην οθόνη ώστε ο χρήστης να προσπαθήσει ξανά. Επίσης ενημερώνεται το εικονίδιο του ImageButton.

Ο κώδικας του Statistics.java, φαίνεται παρακάτω:

```
package com.example.smartalert;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.speech.SpeechRecognizer;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class Statistics extends AppCompatActivity {
    FirebaseDatabase firebaseDatabase;
    DatabaseReference
databaseReference,databaseReference1,databaseReference2,databaseRefer
ence3;
    TextView textView;
    String Fall,Fire,Snow,Flood;
    ImageButton imageButton;
    SpeechRecognizer speechRecognizer;
```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_statistics);
    ImageButton=findViewById(R.id.imageButton);

    speechRecognizer=SpeechRecognizer.createSpeechRecognizer(this);
    textView=findViewById(R.id.textView9);
    firebaseDatabase = FirebaseDatabase.getInstance();
    //get data from Firebase Database
    databaseReference=firebaseDatabase.getReference("countFall");
    databaseReference.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        Fall=String.valueOf(snapshot.getValue());
    }
        @Override
        public void onCancelled(@NonNull DatabaseError error) { }
    });

    databaseReference1=firebaseDatabase.getReference("countFire");
    databaseReference1.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        Fire=String.valueOf(snapshot.getValue());
    }
        @Override
        public void onCancelled(@NonNull DatabaseError error) { }
    });

    databaseReference2=firebaseDatabase.getReference("countSnow");
    databaseReference2.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        Snow=String.valueOf(snapshot.getValue());
    }
        @Override
        public void onCancelled(@NonNull DatabaseError error) { }
    });

    databaseReference3=firebaseDatabase.getReference("countFlood");
    databaseReference3.addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        Flood=String.valueOf(snapshot.getValue());
    }
        @Override
        public void onCancelled(@NonNull DatabaseError error) { }
    });
}

```

```

    }

    //set text to textView

    @SuppressWarnings("SetTextI18n")
    public void Fall(View view){
        textView.setText(getString(R.string.Fall_alert_called)+"
        "+Fall+" "+getString(R.string.times));
    }
    @SuppressWarnings("SetTextI18n")
    public void Fire(View view){
        textView.setText(getString(R.string.Fire_alert_called)+"
        "+Fire+" "+getString(R.string.times));
    }
    @SuppressWarnings("SetTextI18n")
    public void Snow(View view){
        textView.setText(getString(R.string.Snow_alert_called)+"
        "+Snow+" "+getString(R.string.times));
    }
    @SuppressWarnings("SetTextI18n")
    public void Flood(View view){
        textView.setText(getString(R.string.Flood_alert_called)+"
        "+Flood+" "+getString(R.string.times));
    }
    @SuppressWarnings("SetTextI18n")

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
    @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode==888 && resultCode==RESULT_OK){
            //If voice recognized

            ImageButton.setImageDrawable(getDrawable(R.drawable.ic_baseline_mic_o
            ff_24));

            ArrayList<String> results =

            data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
            //show results to screen
            showMessage("Recognized text",results.toString());
            //check if there are any keywords

            if(results.toString().contains("fire")||results.toString().contains("
            φωτιά")||results.toString().contains("incendie")){

            textView.setText(getString(R.string.Fire_alert_called)+" "+Fire+"
            "+getString(R.string.times));
            }
            else
            if(results.toString().contains("flood")||results.toString().contains(
            "πλημμύρα")|| results.toString().contains("inondation")){

            textView.setText(getString(R.string.Flood_alert_called)+" "+Flood+"
            "+getString(R.string.times));
            }
            else
            if(results.toString().contains("Snow")||results.toString().contains("
            χιόνι")||results.toString().contains("neige")) {

            textView.setText(getString(R.string.Snow alert called)+" "+Snow+"

```

```

"+getString(R.string.times));
    }
    else
if (results.toString().contains("Fall") || results.toString().contains("
πίωση") || results.toString().contains("chute")) {

textView.setText(getString(R.string.Fall_alert_called)+" "+Fall+"
"+getString(R.string.times));
    }
    else{
        Toast.makeText(this,"Please try
again...",Toast.LENGTH_SHORT).show();
    }
}
else{
        Toast.makeText(this,"Please try
again...",Toast.LENGTH_SHORT).show();
    }

}

void showMessage(String title,String message){
    new AlertDialog.Builder(this)
        .setTitle(title)
        .setMessage(message)
        .setCancelable(true)
        .show();
}

public void speak(View view) {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED) {
        //request permission
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.RECORD_AUDIO}, 1);
    }

ImageButton.setImageDrawable(getDrawable(R.drawable.ic_baseline_mic_2
4));
    //Intent for Speech Recognition
    Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,"Please say
something!");
    startActivityForResult(intent, 888);
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if(requestCode==1 &&
grantResults[0]!=PackageManager.PERMISSION_GRANTED){
        return;
    }
}

```

```
}
```

➤ [AndroidManifest.xml](#)

Παρακάτω φαίνεται ο κώδικας του αρχείου AndroidManifest.xml, με τα απαραίτητα uses permission:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.smartalert">

    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SmartAlert">
        <activity
            android:name=".Statistics"
            android:exported="false" />
        <!--
            The API key for Google Maps-based APIs is defined as a
string resource.
            (See the file "res/values/google_maps_api.xml").
            Note that the API key is linked to the encryption key
used to sign the APK.
            You need a different API key for each encryption key,
including the release key that is used to
            sign the APK for publishing.
            You can define the keys for the debug and release
targets in src/debug/ and src/release/.
-->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MainActivity3"
            android:exported="false" />
        <activity
            android:name=".MainActivity2"
            android:exported="true" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
```

```

        <action android:name="android.intent.action.MAIN" />

        <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>

```

➤ [strings.xml](#):

- default τιμές:

```

<resources>
    <string name="app_name">SmartAlert</string>
    <string name="welcome"> WELCOME TO SMART ALERT </string>
    <string name="waiting">Waiting for event...</string>
    <string name="alert">ALERT</string>
    <string name="abort">ABORT</string>
    <string name="start">Start</string>
    <string name="send_message">Send message</string>
    <string name="TIMER">TIMER</string>
    <string name="take_a_picture">2.Take a picture</string>
    <string name="title_activity_maps"
translatable="false">MapsActivity</string>
    <string name="ambulanceAndRelatives">Alert message sent to
ambulance and relatives!</string>
    <string name="AlertCancelled">Alert cancelled!</string>
    <string name="imageUploaded">Image uploaded to Storage</string>
    <string name="EverythingsOk">Alert cancelled.Cancellation Sms
sent.</string>
    <string name="see_your_location">1.Click to see your
location</string>
    <string name="steps">FOLLOW THE STEPS BELOW</string>
    <string name="todo" translatable="false">TODO</string>
    <string name="CancelEverythinsOK">Cancel alert! Everything\'s
OK!</string>
    <string name="fire_alert">fire alert</string>
    <string name="snow_alert">snow alert</string>
    <string name="flood_alert">flood alert</string>
    <string name="view_statistics">View statistics</string>
    <string name="show_statistics_for">Show Statistics for:</string>
    <string name="fall_alarm">Fall alarm</string>
    <string name="fire_alarm">Fire alarm</string>
    <string name="snow_alarm">Snow alarm</string>
    <string name="flood_alarm">Flood alarm</string>
    <string name="Fall_alert_called">Fall alert activated</string>
    <string name="times">times</string>
    <string name="Fire_alert_called">Fire alert activated</string>
    <string name="Snow_alert_called">Snow alert activated</string>
    <string name="Flood_alert_called">Flood alert activated</string>
    <string name="imageDeleted">Previous image deleted from
Storage!</string>
    <string name="sent">Alert message sent</string>
    <string name="cancellation_was_successful">Cancellation was
successful</string>
</resources>

```

- Αγγλικά

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name" translatable="false">SmartAlert</string>
    <string name="abort">ABORT</string>
    <string name="start">Start</string>
    <string name="send_message">Send message</string>
    <string name="take_a_picture">Take a picture</string>
    <string name="welcome">WELCOME TO SMART ALERT</string>
    <string name="waiting">Waiting for event...</string>
    <string name="alert">ALERT</string>
    <string name="ambulanceAndRelatives" translatable="false">Alert
message sent to ambulance and relatives!</string>
    <string name="AlertCancelled">Alert cancelled!</string>
    <string name="imageUploaded">Image uploaded to Storage</string>
    <string name="EverythingsOk">Alert cancelled.Cancellation Sms
sent.</string>
    <string name="see_your_location" translatable="false">1.Click to
see your location</string>
    <string name="steps" translatable="false">FOLLOW THE STEPS
BELOW</string>
    <string name="CancelEverythinsOK" translatable="false">Cancel
alert! Everything\'s OK!</string>
    <string name="imageDeleted" translatable="false">Previous image
deleted from Storage!</string>
    <string name="fire_alert" translatable="false">fire
alert</string>
    <string name="snow_alert" translatable="false">snow
alert</string>
    <string name="flood_alert" translatable="false">flood
alert</string>
    <string name="view_statistics" translatable="false">View
statistics</string>
    <string name="show_statistics_for">Show Statistics for:</string>
    <string name="fall_alarm" translatable="false">fall
alarm</string>
    <string name="fire_alarm" translatable="false">Fire
alarm</string>
    <string name="snow_alarm" translatable="false">Snow
alarm</string>
    <string name="flood_alarm" translatable="false">Flood
alarm</string>
    <string name="Fall_alert_called" translatable="false">Fall alert
activated</string>
    <string name="times" translatable="false">times</string>
    <string name="Fire_alert_called" translatable="false">Fire alert
activated</string>
    <string name="Snow_alert_called" translatable="false">Snow alert
activated</string>
    <string name="Flood_alert_called" translatable="false">Flood
alert activated</string>
    <string name="TIMER">TIMER</string>
    <string name="sent">Alert message sent</string>
    <string name="cancellation_was_successful">Cancellation was
successful</string>
</resources>
```

- Ελληνικά

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">ΕΞΥΠΝΟΣ ΣΥΝΑΓΕΡΜΟΣ</string>
    <string name="TIMER">ΧΡΟΝΟΜΕΤΡΟ</string>
    <string name="start" translatable="false">Έναρξη</string>
    <string name="send_message"
translatable="false">Αποστολή</string>
    <string name="abort">ΑΚΥΡΩΣΗ</string>
    <string name="take_a_picture">2.Λήψη φωτογραφίας</string>
    <string name="welcome">ΚΑΛΩΣΗΡΘΑΤΕ ΣΤΟΝ ΕΞΥΠΝΟ ΣΥΝΑΓΕΡΜΟ</string>
    <string name="waiting">ΑΝΑΜΟΝΗ ΣΥΜΒΑΝΤΟΣ.....</string>
    <string name="alert">ΣΥΝΑΓΕΡΜΟΣ</string>
    <string name="ambulanceAndRelatives">Στάλθηκε μήνυμα συναγερμού
στο ασθενοφόρο και στους συγγενείς!</string>
    <string name="AlertCancelled">Ο συναγερμός ακυρώθηκε!</string>
    <string name="imageUploaded">Η εικόνα αποθηκεύτηκε στο
Storage</string>
    <string name="EverythingOk">Ο συναγερμός ακυρώθηκε.Στάλθηκε
μήνυμα ακύρωσης συναγερμού.</string>
    <string name="see_your_location">1.Δείτε την τοποθεσία
σας</string>
    <string name="steps">ΑΚΟΛΟΥΘΗΣΤΕ ΤΑ ΠΑΡΑΚΑΤΩ ΒΗΜΑΤΑ</string>
    <string name="CancelEverythingOK">Άκυρος ο συναγερμός!Όλα
καλά!</string>
    <string name="imageDeleted">Η προηγούμενη φωτογραφία διαγράφηκε
από το Storage</string>
    <string name="fire_alert">Συναγερμός Φωτιάς</string>
    <string name="snow_alert">Συναγερμός Χιονιού</string>
    <string name="flood_alert">Συναγερμός Πλημμύρας</string>
    <string name="view_statistics">Στατιστικά</string>
    <string name="show_statistics_for">Δείξε στατιστικά για:</string>
    <string name="fall_alarm">Συναγερμός Πτώσης</string>
    <string name="fire_alarm">Συναγερμός Φωτιάς</string>
    <string name="snow_alarm">Συναγερμός Χιονιού</string>
    <string name="flood_alarm">Συναγερμός Πλημμύρας</string>
    <string name="Fall_alert_called">Ο Συναγερμός Πτώσης
ενεργοποιήθηκε</string>
    <string name="times">φορές</string>
    <string name="Fire_alert_called">Ο Συναγερμός Φωτιάς
ενεργοποιήθηκε</string>
    <string name="Snow_alert_called">Ο Συναγερμός Χιονιού
ενεργοποιήθηκε</string>
    <string name="Flood_alert_called">Ο Συναγερμός Πλημμύρας
ενεργοποιήθηκε</string>
    <string name="sent">Στάλθηκε μήνυμα συναγερμού!</string>
    <string name="cancellation_was_successful">Η διαδικασία
ακυρώθηκε</string>
</resources>

```

- Γαλλικά

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Alerte intelligente</string>
    <string name="welcome">BIENVENUE À L'ALERTE
INTELLIGENTE</string>
    <string name="waiting">En attente d'événement...</string>
    <string name="alert">"ALERTE"</string>

```

```

<string name="abort">" AVORTER"</string>
<string name="send_message">Envoyer le message</string>
<string name="start">" Démarrer"</string>
<string name="TIMER">MINUTEUR</string>
<string name="take_a_picture">2. Prenez une photo</string>
<string name="ambulanceAndRelatives">Message d\'alerte envoyé à
l\'ambulance et aux proches !</string>
<string name="AlertCancelled">" Alerte annulée !"</string>
<string name="imageUploaded">" Image téléchargée sur
Storage"</string>
<string name="EverythingOk">"L\'alarme a été annulée. Un message
d\'annulation d\'alarme a été envoyé."</string>
<string name="see_your_location" translatable="false">1. Cliquez
pour voir votre position</string>
<string name="steps">" SUIS LES ÉTAPES"</string>
<string name="CancelEverythingOK">" Annuler l\'alerte ! Tout va
bien!"</string>
<string name="imageDeleted">" Image précédente supprimée de
Storage!"</string>
<string name="fire_alert">alerte incendie</string>
<string name="snow_alert">alerte neige</string>
<string name="flood_alert">alerte aux inondations</string>
<string name="view_statistics">" Afficher les
statistiques"</string>
<string name="show_statistics_for">" Afficher les statistiques
pour :"</string>
<string name="fall_alarm">Alarme de chute</string>
<string name="fire_alarm">" Alarme incendie"</string>
<string name="snow_alarm">" Alarme neige"</string>
<string name="flood_alarm">" Alarme d\'inondation"</string>
<string name="Fall_alert_called">" Alerte chute activée"</string>
<string name="times" translatable="false">" fois"</string>
<string name="Fire_alert_called">Alerte incendie activée</string>
<string name="Snow_alert_called">" Alerte neige activée"</string>
<string name="Flood_alert_called">Alerte inondation
activée</string>
<string name="sent">" Message d\'alerte envoyé"</string>
<string name="cancellation_was_successful">L\'annulation a
réussi</string>
</resources>

```

🌐 Παρακολουθήστε τα βίντεο καταγραφής οθόνης που αναδεικνύουν την χρήση της εφαρμογής