

Γεώργιος Τριφύλλης Π14185

trfdoc@gmail.com

Πανεπιστήμιο Πειραιά

Μάθημα: Σύγχρονα Θέματα Τεχνολογίας

Λογισμικού 4ο Έτος

Περιέχεται όλος ο φάκελος της εργασίας καθώς και APK στην διαδρομή

- UnipiAlert\app\build\outputs\apk\debug

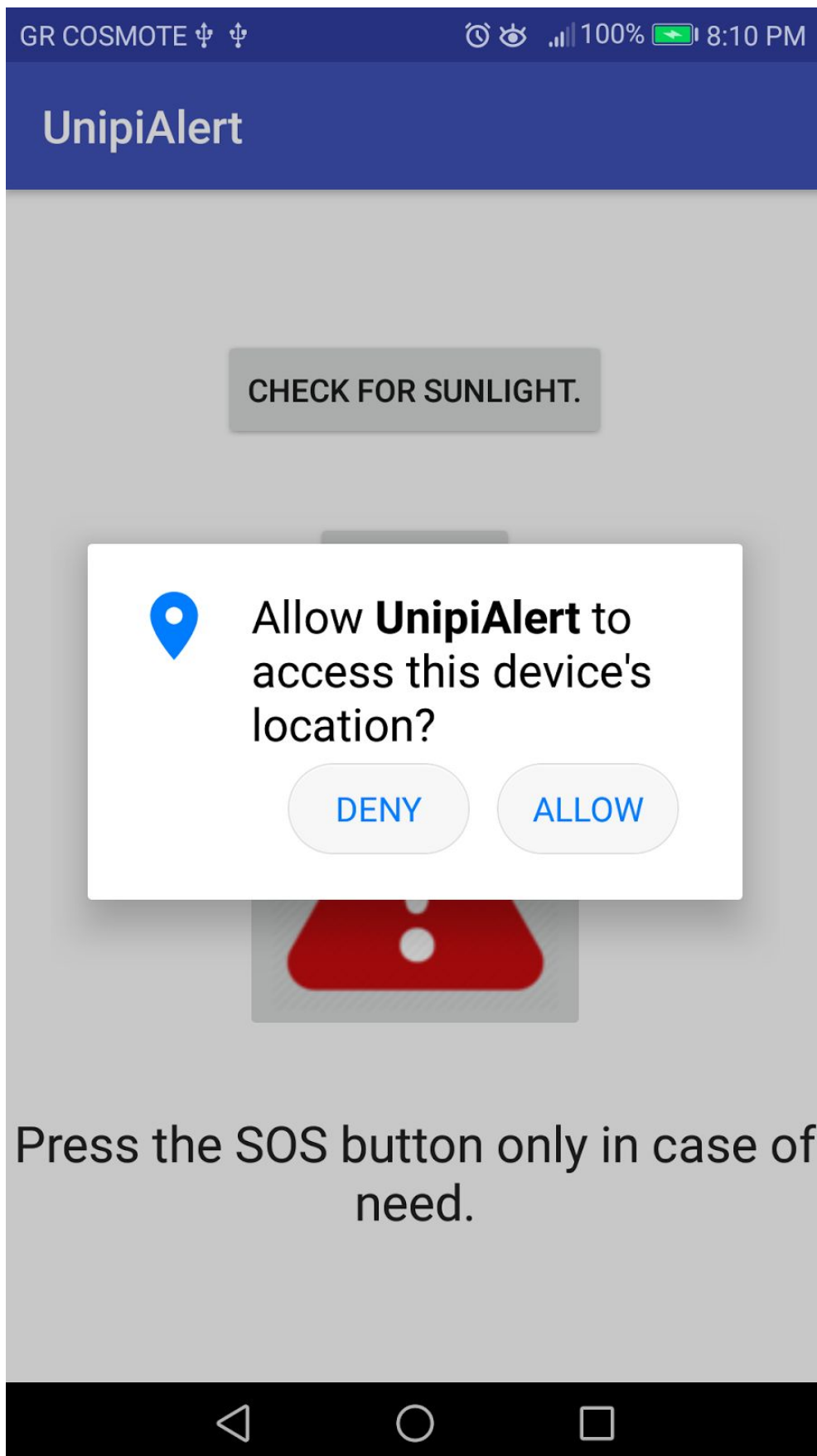
Ο σύνδεσμος για την έρευνα που αναφέρεται μέσα στην εργασία είναι

- <http://www.jcomputers.us/vol19/jcp0907-07.pdf>

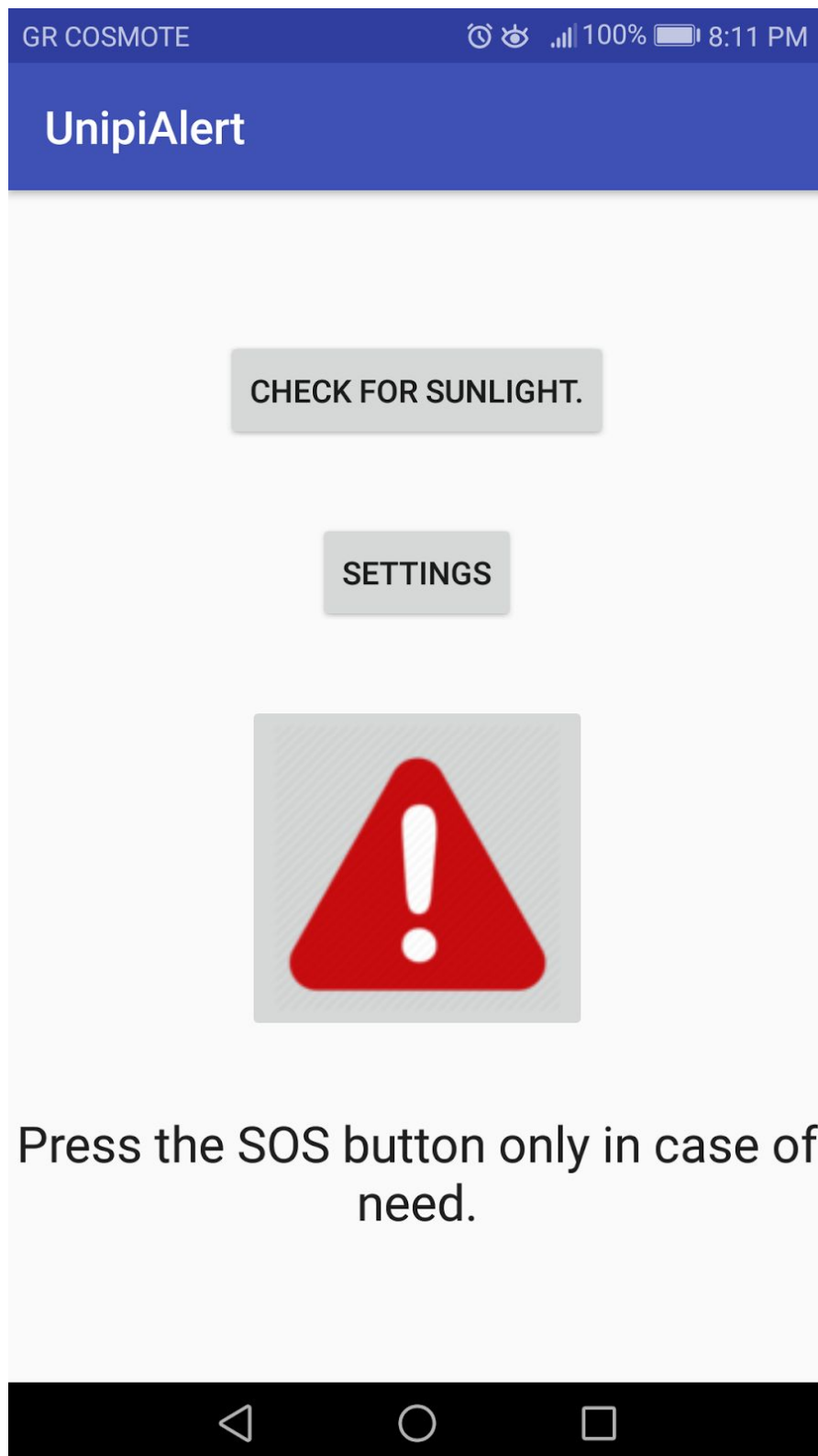
Στο documentation πειέχονται:

- ο πηγαίος κώδικας και η εξήγηση του όπου είναι απαραίτητη
- screenshot από το κινητό καθώς ο επεξεργαστής του υπολογιστή μου δεν μπορεί να υποστηρίξει τον emulator του Android Studio και αυτούς που αναγράφεται ότι μπορεί , “τρέχουν” ανικανοποιητικά σε σημείο που “κρασάρουν” ακόμα και στην αρχική οθόνη του emulator . Προσπάθησα να κατεβάσω άλλους emulator καθώς και προγράμματα για να “τραβάω” μόνο την εικόνα από το κινητό που εξέταζα την εφαρμογή αλλά όλα κατέληξαν σε αποτυχία.

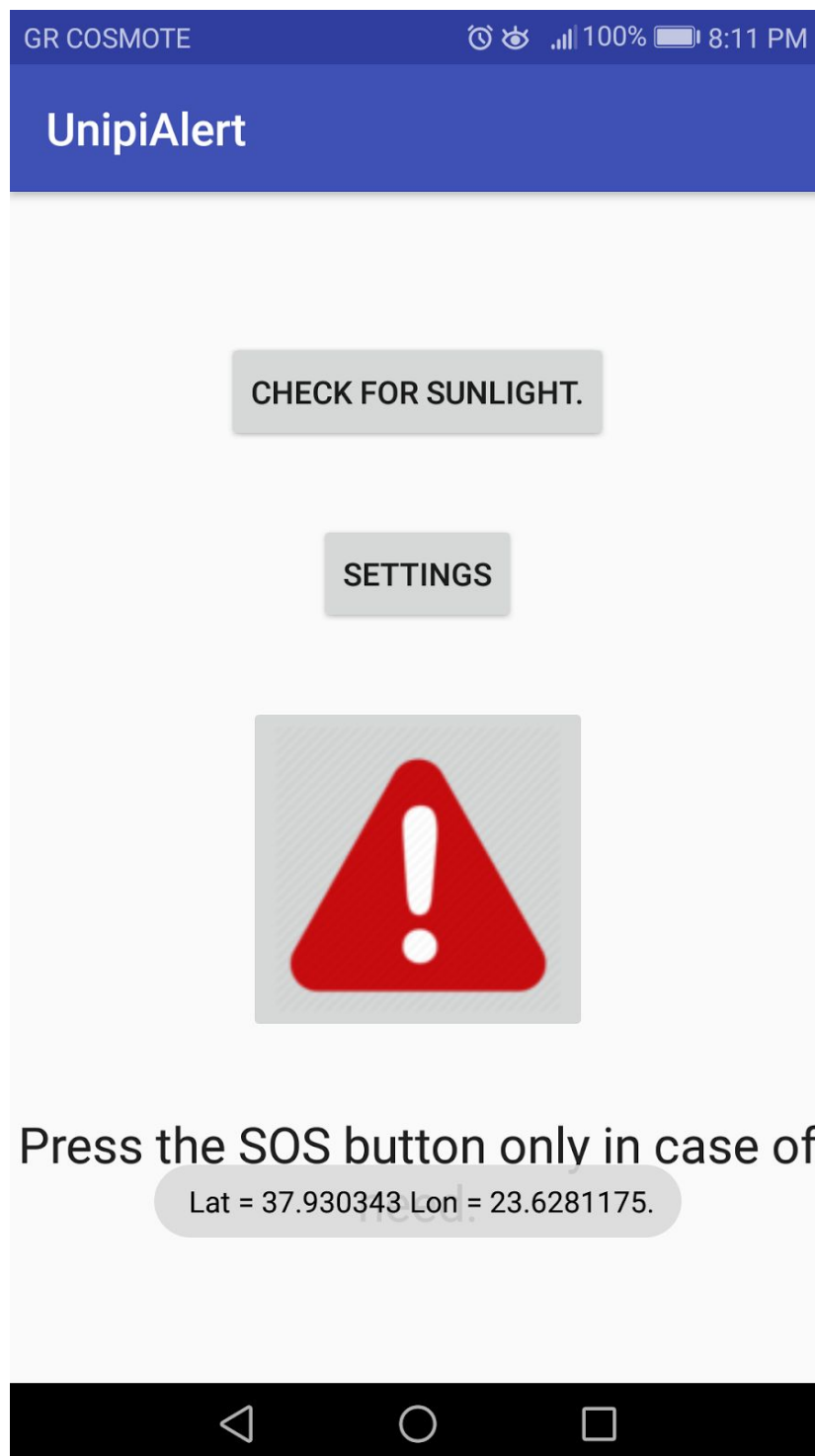
Ερώτηση για εξουσιοδότηση για την χρήση του GPS



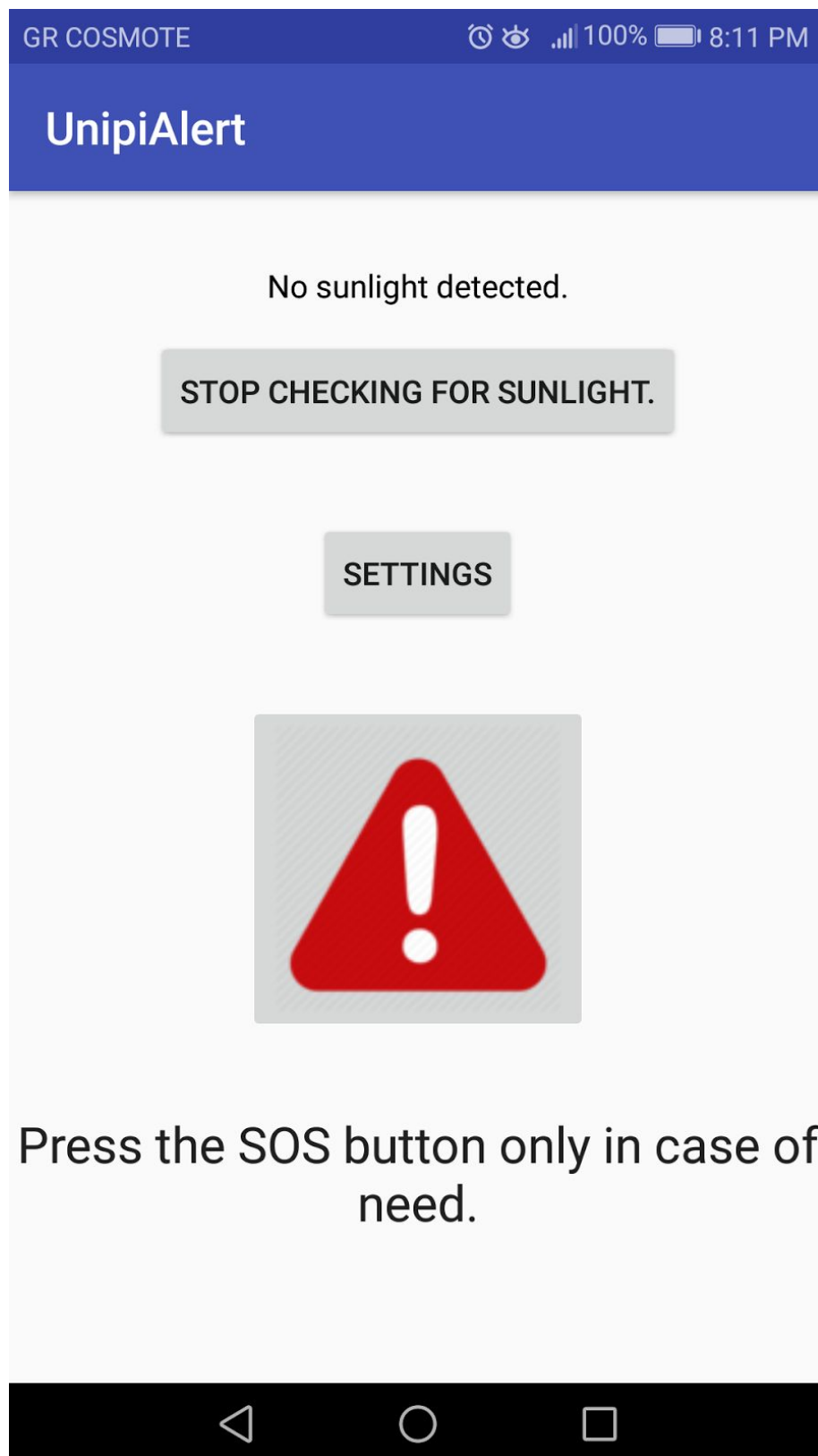
Κεντρική οθόνη της εφαρμογής



Toast μήνυμα για την τοποθεσία του χρήστη

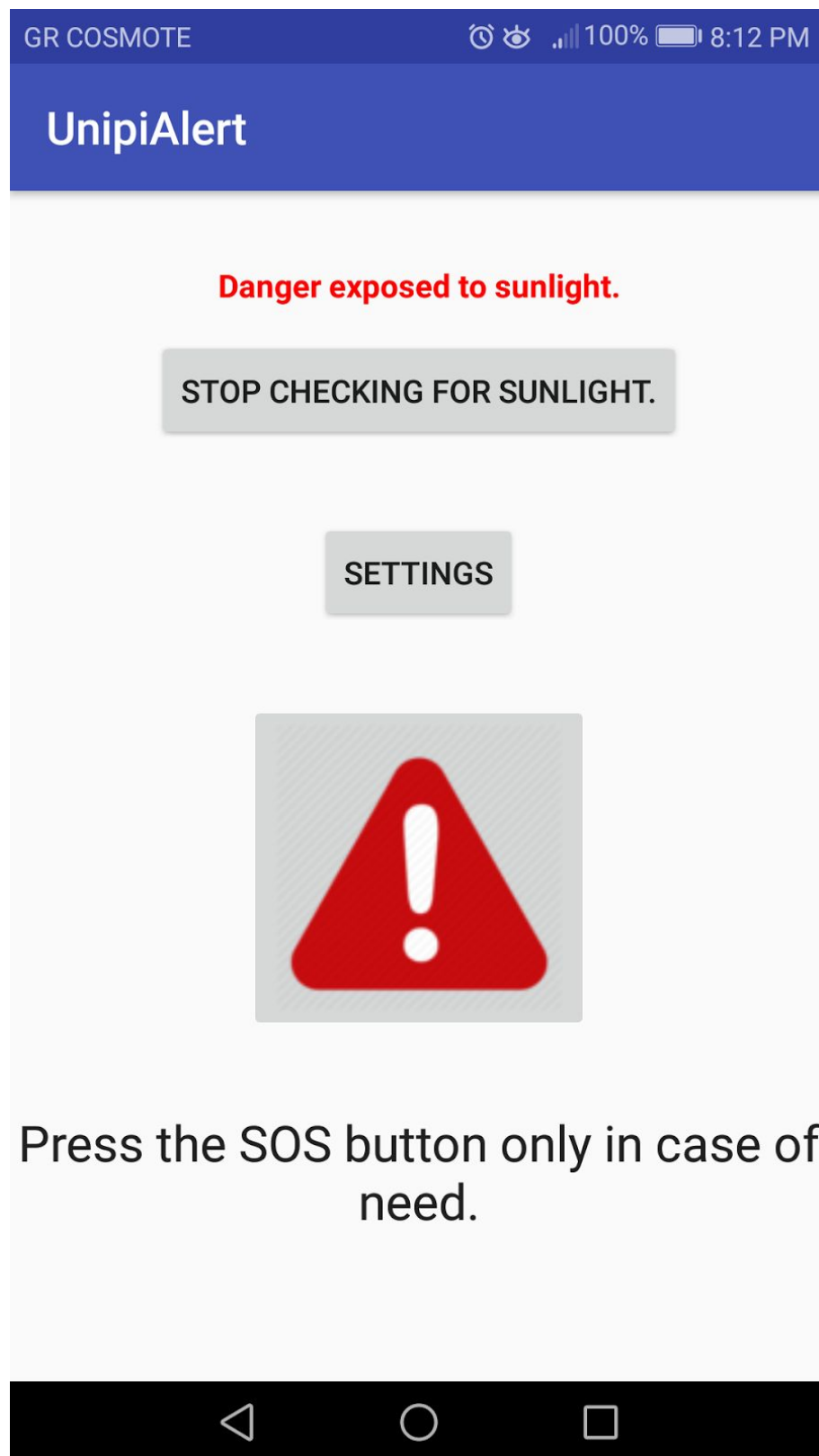


Πάτημα κουμπιού Check For Sunlight.



Τιμή χαμηλότερη της καθορισμένης.

Πάτημα κουμπιού Check For Sunlight.



Τιμή μεγαλύτερη της καθορισμένης

Πάτημα κουμπιού Settings

GR COSMOTE

🕒 👁 📶 100% 🔋 8:12 PM

UnipiAlert

Settings

Username

Password

Recipient Number

Recipient Number

Recipient Number

SAVE!

Αποτυχία συμπλήρωσης των πεδίων των αριθμών

GR COSMOTE 100% 8:12 PM

UnipiAlert

test

Recipient Number

Recipient Number

Recipient Number

SAVE!

Please check the numbers wrong lenght spotted.

English (US)

Activity Abort

GR COSMOTE

🕒 👁 📶 100% 🔋 8:14 PM

UnipiAlert



Abort will send a message to your selected contacts that you are fine and the help message was sent by mistake.



Επιβεβαίωση του Abort

GR COSMOTE

🕒 📶 100% 🔋 8:14 PM

UnipiAlert

Username

Password

CHECK

Please insert your personal
information.



Αποτυχία στον έλεγχο

GR COSMOTE

🕒 👁️ 📶 100% 🔋 8:14 PM

UnipiAlert

Username

Password

CHECK

Please insert your personal
information.

Please enter the right Username and
Password.



Μήνυμα μετά από πτώση

GR COSMOTE

🕒 👁 📶 100% 🔋 8:43 PM

UnipiAlert

CHECK FOR SUNLIGHT.

Please respond

Calling help.

ABORT

Press the SOS button only in case of
need.

Συγγωρέστε με για κάποια screenshot τα οποία δεν υπάρχουν αλλά κάποια πράγματα δεν προλάβαινα να τα τραβήξω καθώς το κινητό δεν έχει συντόμευση για την λήψη screenshot (Toast μετά το πάτημα του κουμπιού Abort καθώς και το Toast για την αποστολή μηνύματος). Όμως σε προηγούμενα screenshot εχώ δώσει παραδείγματα με Toast μηνύματα και θα το δείτε και στα πηγαία αρχεία πως δεν υπάρχει λάθος.

Μετάφραση των βασικών string στα ελληνικά.

Key	Resource Fo...	Untranslata...	Default Value	Greek (el)
app_name	app\src\main	<input checked="" type="checkbox"/>	UnipiAlert	UnipiAlert
Username	app\src\main	<input type="checkbox"/>	Username	Όνομα χρήστη
Password	app\src\main	<input type="checkbox"/>	Password	Κωδικός
Settings	app\src\main	<input type="checkbox"/>	Settings	Ρυθμίσεις
Recipient_Number	app\src\main	<input type="checkbox"/>	Recipient Nu	Αριθμός Πο
Save	app\src\main	<input type="checkbox"/>	Save!	Αποθήκευση
sun_check	app\src\main	<input type="checkbox"/>	Check for sui	Έλεγχος για
sun_stop_check	app\src\main	<input type="checkbox"/>	Stop checkin	Τέλος ελέγ
sun_detected	app\src\main	<input type="checkbox"/>	Danger expo	Κίνδυνος!
sun_not_detected	app\src\main	<input type="checkbox"/>	No sunlight	Δεν εντοπί
check_numbers	app\src\main	<input type="checkbox"/>	Please check	Ελέξτε ότι
Thanks	app\src\main	<input type="checkbox"/>	Thanks	Ευχαριστώ
Sos_cautious	app\src\main	<input type="checkbox"/>	Press the SO	Πατήστε το
Help	app\src\main	<input type="checkbox"/>	Please send t	Παρακαλώ σ
Gps_problem	app\src\main	<input type="checkbox"/>	GPS unable t	To GPS δεν
Abort	app\src\main	<input type="checkbox"/>	Abort	Ακύρωση
Abort_message	app\src\main	<input type="checkbox"/>	Abort will ser	Η ακύρωση
Check_for_abort	app\src\main	<input type="checkbox"/>	Please insert	Παρακαλώ ε
Check	app\src\main	<input type="checkbox"/>	Check	Έλεγχος
wrong_answer	app\src\main	<input type="checkbox"/>	Please enter	Παρακαλώ ε
right_answer	app\src\main	<input type="checkbox"/>	Approved.	Εγκρίθηκε.
help_question	app\src\main	<input type="checkbox"/>	Calling help.	Καλώ για β
help_called	app\src\main	<input type="checkbox"/>	Called for he	Βοήθεια κα
sms_success	app\src\main	<input type="checkbox"/>	SMS Access	Επιβεβαιώε
dialog_title	app\src\main	<input type="checkbox"/>	Please respo	Παρακαλώ ο
NoPrefs	app\src\main	<input checked="" type="checkbox"/>		
Sos_button	app\src\main	<input checked="" type="checkbox"/>	SOS Button	
fall	app\src\main	<input checked="" type="checkbox"/>	Fall	
Abort_SMS	app\src\main	<input type="checkbox"/>	The previous	Το προηγού
Calnceled	app\src\main	<input type="checkbox"/>	Canceled	Ακυρώθηκε
no_support	app\src\main	<input type="checkbox"/>	This device i	Αυτή η συσ
sending_sms	app\src\main	<input type="checkbox"/>	Sending SMS	Αποστέλετο

Κλάση Abort

```
public class Abort extends AppCompatActivity {  
    //Only button handling  
  
    ImageButton BTAbort;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_abort);  
  
        BTAbort = (ImageButton) findViewById(R.id.BTAbort);  
        BTAbort.setOnClickListener((view) -> {  
            Log.e( tag: "textMessage", msg: "AbortPress");  
            gotoActivity(CheckForAbort.class);  
        });  
    }  
  
    private void gotoActivity(Class nextClass){  
        Intent intent = new Intent( packageContext: Abort.this ,nextClass);  
        startActivity(intent);  
    }  
}
```

Η κλάση Abort χρησιμοποιείται μόνο σαν ενδιάμεσο στάδιο και για αυτόν τον λόγο περιέχει μόνο μια αρχικοποίηση κουμπιού και την μέθοδο gotoActivity η οποία λαμβάνει σαν όρισμα την κλάση που θέλουμε να πλοηγηθούμε.

Κλάση CheckAbort

```
public class CheckForAbort extends AppCompatActivity {

    public static final String MyPREFERENCES = "MyPrefs" ;
    public static final String Username = "usernameKey";
    public static final String Password = "passwordKey";
    public static final String Recipient1 = "recipient1Key";
    public static final String Recipient2 = "recipient2Key";
    public static final String Recipient3 = "recipient3Key";

    SharedPreferences ShPref;

    EditText ETPassword , ETUsername;
    Button BTCheck;

    //Ask for permission
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 3;
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        switch (requestCode) {
            case MY_PERMISSIONS_REQUEST_SEND_SMS:
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
                {
                    Toast.makeText(getApplicationContext(), R.string.sms_success , Toast.LENGTH_LONG).show();
                }

                break;
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_check_for_abort);

        BTCheck = (Button) findViewById(R.id.BTCheck);
        ETUsername = (EditText) findViewById(R.id.ETUsername);
        ETPassword = (EditText) findViewById(R.id.ETPassword);

        //If we dont have permission to send a SMS we ask for it
        if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,
                android.Manifest.permission.SEND_SMS)) {
            } else {
                ActivityCompat.requestPermissions( activity: this,
                    new String[] {android.Manifest.permission.SEND_SMS},
                    MY_PERMISSIONS_REQUEST_SEND_SMS);
            }
        }
    }
}
```

Σε αυτό το σημείο μπορούμε να δούμε ότι περνάμε τα κλειδιά και κατά την onCreate κάνουμε τις σωστές συνδέσεις μεταξύ UI και κώδικα . Στην συνέχεια κάνουμε έλεγχο αν έχουμε άδεια να χρησιμοποιήσουμε τα SMS , αν δεν τον έχουμε απλά τον ζητάμε.

```

//Listener that sends SMS when the values are correct
BTCheck.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        SharedPreferences MyPREFERENCES = Context.MODE_PRIVATE);

        String username = ShPref.getString(Username, "");
        String password = ShPref.getString>Password, "");

        String usernameInserted = ETUsername.getText().toString();
        String passwordInserted = ETPassword.getText().toString();

        if(username.equals(usernameInserted) && password.equals(passwordInserted)){
            Log.e( tag: "checkforabort", msg: "CheckRight");
            sendSMS();
            gotoActivity(MainActivity.class);
        }else{
            Toast.makeText(getApplicationContext(), "Please enter the right Username and Password.", Toast.LENGTH_SHORT).show();
        }
    }
});
}

private void sendSMS() {
    if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
        return;
    }

    Long timestampLong = System.currentTimeMillis()/1000;
    String timestamp = timestampLong.toString();
    String timestampKey = timestamp + getResources().getString(R.string.Abort);

    SharedPreferences.Editor editor = ShPref.edit();

    editor.putString(timestampKey, timestamp);
    editor.apply();
    editor.commit();

    String recipient1 = ShPref.getString(Recipient1, "");
    String recipient2 = ShPref.getString(Recipient2, "");
    String recipient3 = ShPref.getString(Recipient3, "");

    String phoneNo1 = "+" + recipient1;
    String phoneNo2 = "+" + recipient2;
    String phoneNo3 = "+" + recipient3;
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(phoneNo1, null, getResources().getString(R.string.Abort_SMS), null, null);
    smsManager.sendTextMessage(phoneNo2, null, getResources().getString(R.string.Abort_SMS), null, null);
    smsManager.sendTextMessage(phoneNo3, null, getResources().getString(R.string.Abort_SMS), null, null);
    Log.e( tag: "textMessage", "Approved.");
    Toast.makeText(getApplicationContext(), "Approved.", Toast.LENGTH_SHORT).show();
}

private void gotoActivity(Class nextClass){
    Intent intent = new Intent( packageContext: CheckForAbort.this ,nextClass);
    startActivity(intent);
}
}

```

Εδώ βλέπουμε πως αναθέτουμε ένα Listener στο κουμπί που αναθέσαμε ο οποίος κατα την κλήση του κάνει έλεγχο να δει αν αυτά που έχει εισάγει ο χρήστης ταιριάζουν με αυτά που είναι αποθηκευμένα. Αν ο έλεγχος είναι επιτυχής βάζουμε ένα Timestamp στα SharedPreferences με μοναδικό κλειδί και στέλνουμε αυτοματοποιημένο μήνυμα στους αποθηκευμένους αριθμούς.

Κλάση Main Activity

```
public class MainActivity extends AppCompatActivity implements SensorEventListener, GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener, LocationListener {

    //Variables used in order to get location
    private LocationRequest mLocationRequest;
    private GoogleApiClient mGoogleApiClient;
    private Location mLastLocation;

    //GPS variables
    private static int UPDATE_INTERVAL = 5000; // SEC
    private static int FATEST_INTERVAL = 3000; // SEC
    private static int DISPLACEMENT = 10; // METERS

    //Shared Preferences
    public static final String MyPREFERENCES = "MyPrefs" ;
    public static final String Recipient1 = "recipient1Key";
    public static final String Recipient2 = "recipient2Key";
    public static final String Recipient3 = "recipient3Key";

    //Drop variables
    private long lastTime = 0;
    private float secondsTillCallingHelp ;

    //Permission variables
    private static final int MY_PERMISSION_REQUEST_CODE = 1;
    private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 2;
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 3;

    TextView textView;
    Button BTCheck_sun , BTSettings ;
    ImageButton BTAlert;

    SharedPreferences ShPref;

    SensorManager sensorManager;
    Sensor lightsensor , accelerationsensor;

    //Checking for SMS and position permissions
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        switch (requestCode) {
            case MY_PERMISSION_REQUEST_CODE:
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    if (checkPlayServices()) {
                        buildGoogleApiClient();
                        createLocationRequest();
                    }
                }

            case MY_PERMISSIONS_REQUEST_SEND_SMS:
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
                {
                    Toast.makeText(getApplicationContext(), R.string.sms_success, Toast.LENGTH_LONG).show();
                }

                break;
        }
    }
}
```

Στην αρχή μπορούμε να δούμε κάποιες μεταβλητές που θα χρειαστούν στην συνέχεια και μια μέθοδο η οποία αναλαμβάνει τις απαντήσεις του χρήστη σε στοιχεία προσβασιμότητας.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Finding buttons
    textView = (TextView) findViewById(R.id.textView);
    BTCheck_sun = (Button) findViewById(R.id.BTCheck_sun);
    BTSettings= (Button) findViewById(R.id.BTSettings);
    BTAlert = (ImageButton) findViewById(R.id.BTAlert);

    ShPref = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);

    //Asking permission during runtime for location
    if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions( activity: this, new String[]{
            android.Manifest.permission.ACCESS_FINE_LOCATION,
            android.Manifest.permission.ACCESS_COARSE_LOCATION
        }, MY_PERMISSION_REQUEST_CODE);
    } else {
        if (checkPlayServices()) {
            buildGoogleApiClient();
            createLocationRequest();
        }
    }

    //Asking permission during runtime for SMS
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale( activity: this,
            Manifest.permission.SEND_SMS)) {
        } else {
            ActivityCompat.requestPermissions( activity: this,
                new String[]{Manifest.permission.SEND_SMS},
                MY_PERMISSIONS_REQUEST_SEND_SMS);
        }
    }

    //Initializing a sensor manager
    sensorManager = (SensorManager) getSystemService(Service.SENSOR_SERVICE);

    //Assigning sensors to sensor manager
    try {
        lightsensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
        accelerationsensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorManager.registerListener( listener: this, lightsensor , SensorManager.SENSOR_DELAY_NORMAL);
        sensorManager.registerListener( listener: this, accelerationsensor , SensorManager.SENSOR_DELAY_NORMAL);
    } catch (Exception e) {
        System.out.println("Error " + e.getMessage());
        return;
    }

    //Listener for BTCheck_sun
    BTCheck_sun.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            changeReadings();
        }
    });
}

```

Στην onCreate βλέπουμε να γίνεται ανάθεση των στοιχείων του UI , έλεγχος για το αν κατέχουμε κατάλληλες προσβάσεις , ανάθεση των sensors καθώς και ένα Listener για το κουμπί BTCheck_sun το οποίο όταν πατηθεί καλεί την changeReadings.

```

//Listener for BTCheck_sun
BTCheck_sun.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        changeReadings();
    }
});

//Listener for BTSettings
BTSettings.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        gotoActivity(Settings.class);
    }
});

//Listener for BTAlert
BTAlert.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Log.e("tag: \"textMessage\", \"msg: \"before\"");
        sendSMS(getResources().getString(R.string.Sos_button));
        gotoActivity(Abort.class);
    }
});
}

@Override
protected void onStart() {
    super.onStart();
    if(mGoogleApiClient!= null)
    {
        mGoogleApiClient.connect();
    }
}

@Override
protected void onPause() {
    super.onPause();
    sensorManager.unregisterListener( listener: this,lightsensor);
    sensorManager.unregisterListener( listener: this,accelerationsensor);
}

@Override
protected void onResume() {
    super.onResume();
    checkPlayServices();
    getLocation();
    sensorManager.registerListener( listener: this,lightsensor,SensorManager.SENSOR_DELAY_NORMAL);
    sensorManager.registerListener( listener: this,accelerationsensor,SensorManager.SENSOR_DELAY_NORMAL);
}

@Override
protected void onStop() {
    LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, locationlistener: this);
    if(mGoogleApiClient != null)
        mGoogleApiClient.disconnect();
    super.onStop();
}
}

```

Εδώ βάζουμε Listener για το κουμπί BTSettings και για το κουμπί BTAlert. Κατά την πάτηση του κουμπιού BTAlert στέλνουμε μηνύματα και στην συνέχεια μεταβαίνουμε στο activity Abort ενώ για το κουμπί BTSettings μεταβαίνουμε στο activity Settings. Στην συνέχεια βλέπουμε τι κάνει η εφαρμογή μας κατά τον κύκλο ζωής της . Κατά την Έναρξη προσπαθούμε να συνδεθούμε με το API της Google το οποίο το χρησιμοποιούμε για την θέση του χρήστη. Κατά την

Παύση αφήνουμε ελεύθερους τους sensors καθώς δεν τους χρησιμοποιούμε και τους αναθέτουμε ξανά κατά το onResume στο οποίο προσπαθούμε να τραβήξουμε την τελευταία τοποθεσία του χρήστη. Ενώ στο onStop σταματάμε και την συνδεσή μας με το API της Google.

```
//Function that sends SMS with a string as an argument in order to create the right timestamp in SharedPreferences
private void sendSMS(String timestampString) {
    //Check for SMS permission
    if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
        return;
    }

    //Getting Location and setting the text message
    String position = getLocation();
    String textMessage = getResources().getString(R.string.Help) + position ;
    Log.e( tag: "position",position);

    //Creating a timestamp
    Long timestampLong = System.currentTimeMillis()/1000;
    String timestamp = timestampLong.toString();
    String timestampKey = timestamp + timestampString ;

    SharedPreferences.Editor editor = ShPref.edit();

    editor.putString(timestampKey, timestamp);
    editor.apply();
    editor.commit();

    String recipient1 = ShPref.getString(Recipient1, " ");
    String recipient2 = ShPref.getString(Recipient2, " ");
    String recipient3 = ShPref.getString(Recipient3, " ");

    String phoneNo1 = "+" + recipient1;
    String phoneNo2 = "+" + recipient2;
    String phoneNo3 = "+" + recipient3;
    SmsManager smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(phoneNo1, sAddress: null, textMessage, sentIntent: null, deliveryIntent: null);
    smsManager.sendTextMessage(phoneNo2, sAddress: null, textMessage, sentIntent: null, deliveryIntent: null);
    smsManager.sendTextMessage(phoneNo3, sAddress: null, textMessage, sentIntent: null, deliveryIntent: null);
    Log.e( tag: "textMessage",textMessage);

    Toast.makeText(getApplicationContext(), R.string.Help_called, Toast.LENGTH_SHORT).show();
}
```

Στην sendSMS πρώτα βλέπουμε αν έχουμε την άδεια να στείλουμε SMS αν την έχουμε συνεχίζουμε αν δεν την έχουμε βγαίνουμε . Στην συνέχεια λαμβάνουμε την τελευταία θέση του χρήστη , ετοιμάζουμε το γραπτό μήνυμα , βάζουμε ένα Timestamp στα SharedPreferences με μοναδικό κλειδί , λαμβάνουμε τους αποθηκευμένους αριθμούς και αφού φτιάξουμε ένα SmsManager τότε αποστέλνουμε τα μηνυματά μας με τους αποθηκευμένους αριθμούς και το

αυτοματοποιημένο μήνυμα. Στο τέλος μεταβιβάζουμε στον χρήστη ανάλογο μήνυμα μέσω Toast.

```
// Function for getting Location
private String getLocation() {
    //Checking for permission
    if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        return null;
    }
    mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
    if (mLastLocation != null) {
        // Splitting Location is Latitude and Longitude
        double latitude = mLastLocation.getLatitude();
        double longitude = mLastLocation.getLongitude();
        //Setting text message
        String textMessage = " Lat = " + latitude + " Lon = " + longitude + ".";
        Log.e( tag: "location", String.valueOf(latitude));
        Log.e( tag: "location", String.valueOf(longitude));
        Toast.makeText(getApplicationContext(), textMessage, Toast.LENGTH_LONG).show();
        return textMessage;
    }
    else
    {
        //Else provide a Toast
        Toast.makeText(getApplicationContext(), R.string.Gps_problem, Toast.LENGTH_SHORT).show();
        return null;
    }
}

//Setting the location request in order to have a precise position
private void createLocationRequest() {
    mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(UPDATE_INTERVAL);
    mLocationRequest.setFastestInterval(FATEST_INTERVAL);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    mLocationRequest.setSmallestDisplacement(DISPLACEMENT);
}

//Setting the API
private synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder( context: this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API).build();
    mGoogleApiClient.connect();
}

//This check is only required if the application is used in a non
private boolean checkPlayServices() {
    int resultCode = GoogleApiAvailability.getInstance().isGooglePlayServicesAvailable( context: this);
    if (resultCode != ConnectionResult.SUCCESS) {
        if (GoogleApiAvailability.getInstance().isUserResolvableError(resultCode)) {
            GoogleApiAvailability.getInstance().getErrorDialog( activity: this, resultCode, PLAY_SERVICES_RESOLUTION_REQUEST).show();
        } else {
            Toast.makeText(getApplicationContext(), R.string.no_support, Toast.LENGTH_LONG).show();
            finish();
        }
        return false;
    }
    return true;
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    getLocation();
}
```

Η getLocation βρίσκει την θέση του χρήστη και την χωρίζει σε γεωγραφικό μήκος και και γυρνάει μία συμβολοσειρά η

οποία περιέχει το αυτοματοποιημένο μήνυμα μας.
Ανεξάρτητα απο την επιτυχία ή μη εμφανίζουμε σχετικό
μήνυμα στον χρήστη μεσω Toast.

```
@Override
public void onConnectionSuspended(int i) {
    mGoogleApiClient.connect();
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {

}

@Override
public void onLocationChanged(Location location) {
    mLastLocation = location;
    getLocation();
}

//Check which Event belongs to what sensor
@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    if (sensorEvent.sensor.getType() == Sensor.TYPE_LIGHT) {
        //If it is the light one , then change the TextView accordingly
        if (sensorEvent.values[0] > 50) {
            textView.setTextColor(Color.RED);
            textView.setTypeface(Typeface.DEFAULT_BOLD);
            textView.setText(R.string.sun_detected);
        } else {
            textView.setTextColor(Color.BLACK);
            textView.setTypeface(Typeface.DEFAULT);
            textView.setText(R.string.sun_not_detected);
        }
    } else if (sensorEvent.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        //if it is the accelerometer then make the math below
        //http://www.jcomputers.us/vol9/jcp0907-07.pdf
        //The "impact shock" results in an intense inversion of the polarity of the acceleration.
        //TABLE I THRESHOLDS FOR DIFFERENT POSITIONS
        //Threshold for shock is 2.02g, for this reason I compare the rootSquare with 2.0(g)

        float x = sensorEvent.values[0];
        float y = sensorEvent.values[1];
        float z = sensorEvent.values[2];
        double rootSquare = 0.0;

        long curTime = System.currentTimeMillis();

        //From the study we found out that a typical fall is about 800ms so we check every 200ms for the impact shock
        if ((curTime - lastTime) > 200) {
            lastTime = curTime;
            //Check the provided journal
            rootSquare = Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2) + Math.pow(z, 2));
            if (rootSquare < 2.0) {
                timerForSendingSMS();
            }
        }
    }
}
```

Στον onSensorChanged αρχικά βλέπουμε ποιός απο τους δύο
sensor που έχουμε προκάλεσε το γεγονός. Αν αυτός ήταν του

φωτός τότε λαμβάνουμε την τιμή του και τροποποιούμε το `textView` αναλόγως(η τιμή 50 είναι ενδεικτική) . Αν ήταν ο `sensor` της επιτάχυνσης τότε κάθε 200 ms συγκρίνουμε τις τιμές του. Περισσότερες πληροφορίες σχετικά με την μαθηματική εξίσωση μπορείτε να βρείτε στον σύνδεσμο της πρώτης σελίδας . Ο σύνδεσμος επίσης παρέχεται και μέσα στον πηγαίο κώδικα . Αν ικανοποιηθεί η συνθήκη καλούμε την `timerForSendingSMS`.

```

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

//function for changing the button string
public void changeReadings() {
    if(textView.getVisibility() == View.VISIBLE){
        BTCheck_sun.setText("Check for sunlight.");
        textView.setVisibility(View.INVISIBLE);
    }else{
        BTCheck_sun.setText("Stop checking for sunlight.");
        textView.setVisibility(View.VISIBLE);
    }
}

//Function that helps us navigate through our app
private void gotoActivity(Class nextClass){
    Intent intent = new Intent( packageContext: MainActivity.this ,nextClass);
    startActivity(intent);
}

//Function for sending message after 30 seconds
public void timerForSendingSMS() {

    // Setting a Dialog
    final AlertDialog.Builder timerDialog = new AlertDialog.Builder( context: this);
    timerDialog.setTitle(getString(R.string.dialog_title));
    timerDialog.setMessage(getString(R.string.help_question));
    // Providing sound so that we can attract user's attention
    final MediaPlayer mp = MediaPlayer.create( context: this, R.raw.clock_tac);

    //Initiating a 30 second timer with a 1 second step
    final CountDownTimer timer = new CountDownTimer( millisInFuture: 30000, countDownInterval: 1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            secondsTillCallingHelp = (millisUntilFinished / 100);
            //Play sound
            mp.start();
            // Tried to refresh the message of the Dialog but I couldn't get it to work that way.
            // timerDialog.setMessage(getString(R.string.help_question) + secondsTillCallingHelp);
        }

        @Override
        public void onFinish() {
            timerDialog.setMessage(getString(R.string.help_called));
            sendSMS(getResources().getString(R.string.fall));
            //Announcing to user that the messages are sent.
            Toast.makeText(getApplicationContext(), R.string.sending_sms, Toast.LENGTH_SHORT).show();
        }
    };

    timerDialog.setCancelable(false);
    timerDialog.setPositiveButton( text: "Abort", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface aboutDialog, int id) {
            //Announcing to user that he canceled the operation
            Toast.makeText(getApplicationContext(),R.string.Canceled, Toast.LENGTH_SHORT).show();
            //Timer is canceled
            timer.cancel();
        }
    });
    AlertDialog alertDialog = timerDialog.create();
    alertDialog.show();
    timer.start();
}

```

Πρώτα βλέπουμε την `changeReadings` η οποία ενεργοποιεί και απενεργοποιεί το `textView` ενώ παράλληλα αλλάζει και την συμβολοσειρά που εμφανίζεται πάνω στο κουμπί.

Στην μέθοδο `timerForSendingSMS` βλέπουμε πώς φτιάχνουμε ένα `Dialog` ένα `MediaPlayer` και ένα `CountDownTimer`. Το `Dialog` έχει μόνο ένα κουμπί το οποίο κατα την πάτηση του καλεί την `timer.cancel` και ακυρώνει τον `CountDownTimer`. Ο `CountDownTimer` ξεκινάει από τα 30 δευτερόλεπτα και κάθε ένα δευτερόλεπτο παίζει ένα αρχείο ήχου με το οποίο προσπαθούμε να τραβήξουμε την προσοχή του χρήστη.

Κλάση Settings

```
public class Settings extends AppCompatActivity {

    EditText ETUsername,ETPassword,ETRecipient1,ETRecipient2,ETRecipient3;
    Button BTSave;
    SharedPreferences ShPref;

    public static final String MyPREFERENCES = "MyPrefs" ;
    public static final String Username = "usernameKey";
    public static final String Password = "passwordKey";
    public static final String Recipient1 = "recipient1Key";
    public static final String Recipient2 = "recipient2Key";
    public static final String Recipient3 = "recipient3Key";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        ETUsername = (EditText) findViewById(R.id.ETUsername);
        ETPassword = (EditText) findViewById(R.id.ETPassword);
        ETRecipient1 = (EditText) findViewById(R.id.ETRecipient1);
        ETRecipient2 = (EditText) findViewById(R.id.ETRecipient2);
        ETRecipient3 = (EditText) findViewById(R.id.ETRecipient3);

        ShPref = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);

        String username = ShPref.getString(Username, " ");
        String password = ShPref.getString(Password, " ");
        String recipient1 = ShPref.getString(Recipient1, " ");
        String recipient2 = ShPref.getString(Recipient2, " ");
        String recipient3 = ShPref.getString(Recipient3, " ");

        ETUsername.setText(username);
        ETPassword.setText(password);
        ETRecipient1.setText(recipient1);
        ETRecipient2.setText(recipient2);
        ETRecipient3.setText(recipient3);

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

        BTSave.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                if(ETRecipient1.length() == 12 && ETRecipient2.length() == 12 && ETRecipient3.length() == 12 )
                // check for 12 because the global prefix is 2 numbers
                {
                    String username = ETUsername.getText().toString();
                    String password = ETPassword.getText().toString();
                    String recipient1 = ETRecipient1.getText().toString();
                    String recipient2 = ETRecipient2.getText().toString();
                    String recipient3 = ETRecipient3.getText().toString();
                    Long timestampLong = System.currentTimeMillis()/1000;
                    String timestamp = timestampLong.toString();
                    String timestampKey = timestamp + "Settings" ;

                    //Saving values from the EditTexts to the SharedPreferences.

                    SharedPreferences.Editor editor = ShPref.edit();

                    editor.putString(timestampKey, timestamp);
                    editor.putString(Username, username);
                    editor.putString(Password, password);
                    editor.putString(Recipient1, recipient1);
                    editor.putString(Recipient2, recipient2);
                    editor.putString(Recipient3, recipient3);
                    editor.apply();
                    editor.commit();
                    Toast.makeText( Context.Settings.this, "Thanks", Toast.LENGTH_LONG).show();
                    finish();
                }
                else{
                    Toast.makeText( Context.Settings.this, "Please check the numbers wrong lenght spotted.", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

Η κλάση Settings απλά τραβαί και αποθηκεύει δεδομένα στα SharedPreferences.

Όλα τα XML αρχεία τα οποία αφορούν το UI έγιναν με ScrollView το οποίο περιέχει RelativeLayout το οποίο περιέχει όλα τα στοιχεία που εμφανίζονται . Επίσης έχει προστεθεί εξάρτηση στο Play Service το οποίο χρειάζεται για την ομαλή διεξαγωγή της εντόπισης θέσης.

Αν υπάρξει οποιοδήποτε πρόβλημα θα παρακαλούσα να επικοινωνήσετε μαζί μου στο email που δίνεται στο εξώφυλλο.