



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Σχεδίαση και ανάπτυξη εφαρμογής Android αναγνώρισης πτώσεων</b>  <b>Design and development of an Android Application for fall detection</b>
Όνοματεπώνυμο Φοιτητή	<b>ΚΑΠΕΤΑΝΙΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ</b>
Πατρώνυμο	<b>ΔΗΜΗΤΡΙΟΣ</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ15022</b>
Επιβλέπων	<b>ΑΛΕΠΗΣ ΕΥΘΥΜΙΟΣ</b>

Ημερομηνία Παράδοσης **Μάρτιος 2018**

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Αλέπης Ευθύμιος  
Επίκουρος Καθηγητής

Πασάκης Κωνσταντίνος  
Επίκουρος Καθηγητής

Τσιχριντζής Γεώργιος  
Καθηγητής

## **Ευχαριστίες**

Ευχαριστώ όλους τους ανθρώπους που με στήριξαν στην απόφασή μου να ξεκινήσω το μεταπτυχιακό αυτό και κυρίως την σύντροφό μου Γαρυφαλλιά και τους γονείς μου, οι οποίοι με βοήθησαν σε όλα τα στάδια με τη στάση τους.

Ευχαριστώ τον φίλο, συμφοιτητή και συνεργάτη Ανδρέα Κουρεμάδα, για την υπομονή, τις ατελείωτες ώρες δουλειάς του και την υπέροχη συνεργασία που είχαμε κατά τη διάρκεια της εκπόνησης της διατριβής μας.

Ευχαριστώ τους καθηγητές μου από το Πολυτεχνείο Κρήτης, κύριο Βασίλη Κουϊκόγλου και κύριο Δημήτρη Πατέλη που με παροτρύναν και με βοήθησαν, ο καθένας με τον τρόπο του, να ακολουθήσω το δρόμο που πραγματικά ήθελα.

Τέλος, οφείλω ένα μεγάλο ευχαριστώ στους καθηγητές μου, κύριο Ε. Αλέπη και Κ. Πατσάκη για τις γνώσεις που μου μετέδωσαν, την άψογη συνεργασία και την ευκαιρία που έδωσαν στον Ανδρέα και έμένα, ώστε να αναπτύξουμε μαζί ένα θέμα που πραγματικά μας ενδιέφερε.

## Περίληψη

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η σχεδίαση και ανάπτυξη ενός συστήματος ανίχνευσης πτώσεων και άμεσης παροχής πρώτων βοηθειών. Η διατριβή έγινε σε συνεργασία με τον Ανδρέα Κουρεμάδα (ΜΠΠΛ 15032), ο οποίος ανέπτυξε τη διαδικτυακή εφαρμογή για την ενημέρωση και απεικόνιση των δεδομένων.

Το κομμάτι του συστήματος που παρουσιάζεται εδώ, αφορά την εφαρμογή που είναι υπεύθυνη για την ανίχνευση της πτώσης. Η ανάπτυξή της έγινε για έξυπνες συσκευές (κινητά και tablets) με λειτουργικό android. Η ανίχνευση της πτώσης γίνεται μέσω των δεδομένων της επιτάχυνσης που μας παρέχει η συσκευή από το επιταχυνσιόμετρο και αποστέλλονται σε κεντρική βάση δεδομένων μαζί με τις συντεταγμένες της συσκευής του χρήστη.

Το backend του συστήματος παρουσιάζεται στην διπλωματική εργασία του Α.Κουρεμάδα {Σχεδίαση και ανάπτυξη διαδικτυακής εφαρμογής ενημέρωσης και διαχείρισης πτώσεων}.

## Abstract

In this thesis we are developing an automated system that detects falls and inform the healthcare system to send help. For this reason, we developed two applications, an Android App and a web App.

The Android App is responsible for the fall detection and requesting for help. The detection happens via the use of the data provided by the mobile devices (cell phones and tablets) accelerometer. After a fall has been detected we send the location data to a centralized database.

The systems backend can be found in A. Kouremadas thesis {Design and development of a web-based application for information and management of falls}

## ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες.....	3
Περίληψη.....	4
Abstract.....	4
1. Εισαγωγή.....	6
2. Ανασκόπηση Πεδίου.....	7
2.1 Εφαρμογές.....	7
2.1.1 Fall Detection, Deskshare, Inc.....	7
2.1.2 Fade: fall detector, ITER S.A.....	8
2.1.3 Emergency Fall Detector, Socaplaya21.....	9
2.2 Ερευνητικές Εργασίες.....	9
2.2.1 A new fall detection system on Android smartphone: Application to a SDN-based IoT system	9
2.2.2 Analysis of Android Device-Based Solutions for Fall Detection	10
3. Αρχιτεκτονική Συστήματος.....	11
3.1 Πλατφόρμα υλοποίησης Android application.....	11
3.2 Ανάλυση υλοποίησης Android application.....	11
3.2.1 Android Location API.....	11
3.2.2 Επιταχυνσιόμετρο.....	11
3.2.3 Βιβλιοθήκες εφαρμογής.....	11
3.2.4 Αρχιτεκτονική android application.....	12
3.3 Διαγράμματα περιπτώσεων χρήσης.....	17
4. Εγχειρίδιο Χρήστη.....	21
4.1 Σύντομη παρουσίαση Android Application.....	21
5. Συμπεράσματα και μελλοντικές επεκτάσεις.....	24
6. Βιβλιογραφία.....	25
7. Παράρτημα.....	26
7.1 Κατανομή αγοράς λειτουργικών Android.....	26
7.2 Κώδικας αναγνώρισης πτώσεων.....	27
7.3 Κώδικας χρονοδιακόπτη.....	28
7.4 Permissions Εφαρμογής και λοιπά attributes στο Manifest.....	28
7.5 Μερίδιο της αγοράς (λειτουργικά συστήματα).....	29

## 1. Εισαγωγή

Οι πτώσεις είναι ένα πολύ σημαντικό πρόβλημα που καλείται να αντιμετωπίσει το σύστημα υγείας. Πρέπει να σημειωθεί ότι μπορεί να είναι αποτέλεσμα πολλών διαφορετικών παραγόντων (πχ. έμφραγμα, εγκεφαλικό, απλή πτώση). Η ανίχνευση και η έγκαιρη παροχή βοήθειας στον παθόντα είναι ένα πρόβλημα στο οποίο δεν έχει δοθεί η απαραίτητη προσοχή από το σύστημα υγείας. Μέχρι σήμερα, τα αιτήματα παροχής βοήθειας γίνονται μέσω τρίτων, με αποτέλεσμα να αυξάνεται ο χρόνος ανταπόκρισης στο εκάστοτε περιστατικό.

Αν μπορούσαμε να έχουμε ένα αυτόματο σύστημα ανίχνευσης και ενημέρωσης γι' αυτά τα περιστατικά, ώστε να επιτευχθεί η παροχή βοήθειας, θα κερδίζαμε πολύτιμο χρόνο. Αυτό το πρόβλημα προσπαθήσαμε να επιλύσουμε με την παρούσα μεταπτυχιακή διατριβή.

Για να το επιτύχουμε, δημιουργήσαμε μια εφαρμογή (app) για κινητές συσκευές με λειτουργικό android, η οποία ανιχνεύει πτώσεις και ενημερώνει αυτόματα μια κεντρική βάση. Επιπλέον, η εφαρμογή μας δίνει τη δυνατότητα στο χρήστη να ζητήσει ιατρική βοήθεια με το πάτημα ενός κουμπιού από την επιφάνεια εργασίας του, αν θεωρήσει ότι είναι αναγκαίο.

Τα δεδομένα που αποστέλλει το android app, καταγράφονται σε μια βάση δεδομένων και είναι τα παρακάτω:

- Ονοματεπώνυμο παθόντα
- Ημερομηνία γέννησης
- Πιθανές ασθένειες από τις οποίες πάσχει ο χρήστης
- Τοποθεσία περιστατικού
- Android ID

Το σύστημά μας είναι σχεδιασμένο με τέτοιο τρόπο, ώστε να καλύπτει το μεγαλύτερο εύρος έξυπνων συσκευών.

Το android app έχει τη δυνατότητα να λειτουργεί συνεχόμενα, ακόμη και αν η συσκευή είναι κλειδωμένη μέσω ενός service.

Επίσης, δίνεται η δυνατότητα στο χρήστη να ακυρώσει το αίτημα βοήθειας εντός εύλογου χρονικού διαστήματος (30 δευτερόλεπτα).

Η επιλογή του λειτουργικού συστήματος Android έγινε με βάση το ποσοστό της αγοράς που κατέχει. Σύμφωνα με στατιστικές (Παράρτημα 7.5) το Android κατείχε το 87.7% της αγοράς το δεύτερο τετράμηνο του 2017. Δεδομένου ότι τα τελευταία χρόνια έχει σταθερά ποσοστό μεγαλύτερο του 75%, θεωρήσαμε πως η ανάπτυξη της εφαρμογής έπρεπε να γίνει για Android συσκευές.

## 2. Ανασκόπηση Πεδίου

Στο Play Store υπάρχουν οι κάτωθι παρόμοιες εφαρμογές:

- Fall Detection, Deskshare, Inc
- Fade: fall detector, ITER S.A.
- Emergency Fall Detector, Socaplaya21

οι οποίες δυστυχώς δεν απαντούν στο πρόβλημα που τέθηκε στην εισαγωγή, δηλαδή την ενημέρωση του συστήματος υγείας μέσω τρίτων. Πρέπει να σημειωθεί ότι υπάρχουν κάποιες δημοσιεύσεις, οι οποίες όμως αποσκοπούν στη βελτίωση της ανίχνευσης πτώσεων και όχι στην άμεση ενημέρωση του συστήματος υγείας.

### 2.1 Εφαρμογές

#### 2.1.1 Fall Detection, Deskshare, Inc

Η συγκεκριμένη εφαρμογή παρέχει:

- Υπηρεσία αναγνώρισης πτώσης
- Αποστολή γραπτού μηνύματος (SMS) σε τρίτο πρόσωπο για το συμβάν
- Δυνατότητα ακύρωσης της αποστολής του SMS
- Δυνατότητα αλλαγής της ευαισθησίας της αναγνώρισης πτώσης



Εικόνα 1. Στιγμιότυπα από την εφαρμογή Fall Detection (Deskshare, Inc)

Η εφαρμογή Fall Detection (Deskshare, Inc) διατίθεται στο Play Store έναντι 3,19€ μέσω της ακόλουθης διεύθυνσης:

[https://play.google.com/store/apps/details?id=com.fall\\_detection&hl=en](https://play.google.com/store/apps/details?id=com.fall_detection&hl=en)

### 2.1.2 Fade: fall detector, ITER S.A.

Η συγκεκριμένη εφαρμογή παρέχει:

- Υπηρεσία αναγνώρισης πτώσης
- Αποστολή γραπτού μηνύματος (SMS και email) σε τρίτο πρόσωπο για το συμβάν
- Αυτόματη τηλεφωνική κλήση σε επαφή του χρήστη με ενεργοποίηση ανοιχτής ακρόασης

Η εφαρμογή αυτή βρίσκεται ακόμη στο στάδιο των ελέγχων.



Εικόνα 2. Στιγμιότυπα από την εφαρμογή Fade: fall detector (ITER S.A)

Η εφαρμογή Fade: fall detector (ITER S.A.) διατίθεται στο Play Store δωρεάν μέσω της ακόλουθης διεύθυνσης:

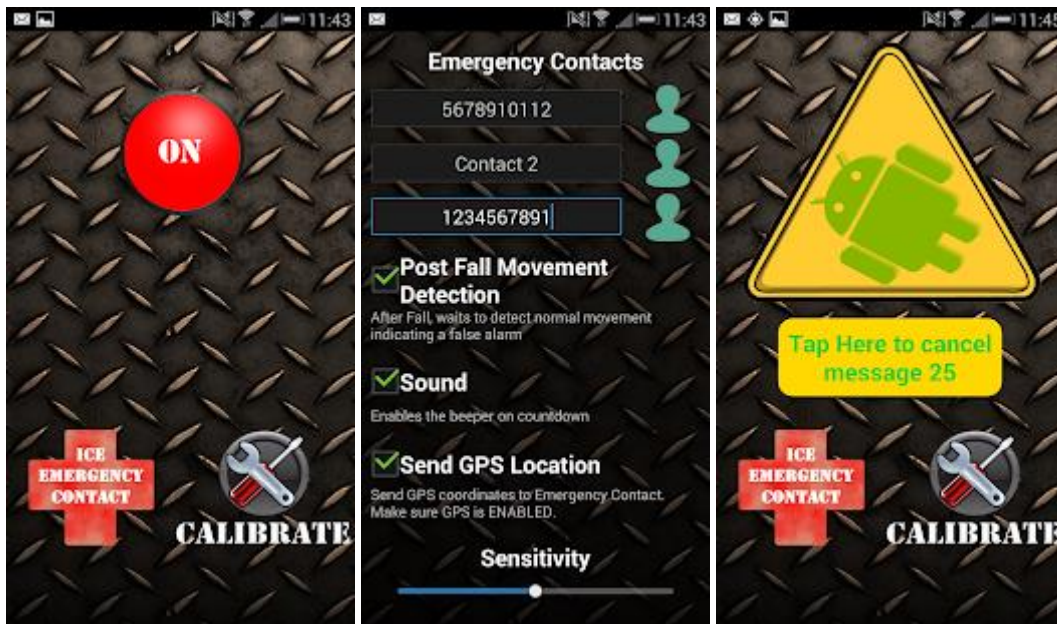
<https://play.google.com/store/apps/details?id=com.iter.falldetector&hl=en>



### 2.1.3 Emergency Fall Detector, Socaplaya21

Η συγκεκριμένη εφαρμογή παρέχει:

- Υπηρεσία αναγνώρισης πτώσης
- Αποστολή γραπτού μηνύματος (SMS) σε τρίτο πρόσωπο για το συμβάν
- Δυνατότητα αλλαγής της ευαισθησίας της αναγνώρισης πτώσης



Εικόνα 3. Στιγμιότυπα από την εφαρμογή Emergency Fall Detector (Socaplaya21)

Η εφαρμογή Emergency Fall Detector (Socaplaya21) διατίθεται στο Play Store δωρεάν μέσω της ακόλουθης διεύθυνσης:

<https://play.google.com/store/apps/details?id=soca.fallprotection&hl=en>

## 2.2 Ερευνητικές Εργασίες

### 2.2.1 A new fall detection system on Android smartphone: Application to a SDN-based IoT system

Hai Anh Tran, Quynh Thu Ngo, Van Tong

Η παραπάνω ερευνητική εργασία έχει ως σκοπό την ανάπτυξη μιας εφαρμογής Android για αναγνώριση πτώσης κάνοντας χρήση machine learning και data mining. Το κύριο πρόβλημα που προσπαθεί να λύσει είναι η ακρίβεια στην αναγνώριση και όχι η άμεση παροχή βοήθειας που είναι ο δικός μας στόχος.

Η εργασία βρίσκεται στον παρακάτω σύνδεσμο:

<http://ieeexplore.ieee.org/document/8119425/>

Σχεδίαση και ανάπτυξη εφαρμογής Android αναγνώρισης πτώσεων

### **2.2.2 Analysis of Android Device-Based Solutions for Fall Detection**

Eduardo Casilari,\* Rafael Luque, and María-José Morón

Ki H. Chon, Academic Editor

Η παραπάνω ερευνητική εργασία έχει ως σκοπό την κατηγοριοποίηση και αξιολόγηση των διάφορων γνωστών μεθοδολογιών αναγνώρισης πτώσεων.

Αρχικά, πραγματεύεται μια γενική κατηγοριοποίηση των μεθοδολογιών δίνοντας ιδιαίτερη σημασία στις συσκευές Android και συγκεκριμένα στα smartphones. Ακολουθώς, αναλύει λεπτομερώς τις υπάρχουσες ερευνητικές εργασίες που βασίζονται σε Android, από διαφορετικές σκοπιές. Πιο συγκεκριμένα, οι σκοπιές αυτές είναι:

- Αρχιτεκτονική συστήματος
- Τυπολογία και παραμετροποίηση του αλγόριθμου που κάνει την αναγνώριση
- Τρόποι αντίδρασης του εκάστοτε συστήματος στην αναγνώριση πτώσης
- Αποτελεσματικότητα του εκάστοτε συστήματος

Η εργασία βρίσκεται στον παρακάτω σύνδεσμο:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4570297/>

### 3. Αρχιτεκτονική Συστήματος

#### 3.1 Πλατφόρμα υλοποίησης Android application

Η ανάπτυξη αυτού του υποσυστήματος έγινε στο λειτουργικό Android χρησιμοποιώντας την τελευταία έκδοση του ολοκληρωμένου προγραμματιστικού περιβάλλοντος (IDE) Android Studio.

Το Android Studio είναι βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, είναι σχεδιασμένο αποκλειστικά για προγραμματισμό Android (Ducrohet & Norbye & Chou, 2013) και αποτελεί το κύριο ολοκληρωμένο προγραμματιστικό περιβάλλον της Google για ανάπτυξη εφαρμογών Android. Είναι διαθέσιμο για Windows, Linux και Mac OS X και διατίθεται ελεύθερα προς χρήση.

Η γλώσσα ανάπτυξης που χρησιμοποιήθηκε είναι Java, αντικειμενοστρεφής γλώσσα προγραμματισμού, που σχεδιάστηκε από την εταιρία πληροφορικής Sun Microsystems και πρωτοεμφανίστηκε πριν από 22 χρόνια, στις 23 Μαΐου του 1995 (Binstock, 2015).

#### 3.2 Ανάλυση υλοποίησης Android application

##### 3.2.1 Android Location API

Το Location API του Android χρησιμοποιείται στην εφαρμογή μας για τον εντοπισμό του σημείου της πτώσης. Μέσω της τάξης «LocationManager», η οποία παρέχεται από το πακέτο «android.location» και μας επιτρέπει να λαμβάνουμε περιοδικές ενημερώσεις για την τοποθεσία της συσκευής, Αυτό γίνεται με τη χρήση ενός αντικειμένου τύπου «LocationListener» («android.location.LocationListener»).

##### 3.2.2 Επιταχυνσιόμετρο

Μέσω των δεδομένων που παρέχονται από τον αισθητήρα επιτάχυνσης της συσκευής μπορούμε με σχετική ακρίβεια να αναγνωρίσουμε αν υπάρχει πτώση. Τα δεδομένα συλλέγονται μέσω των τάξεων «SensorManager» και «SensorEventListener», η πρώτη με τις κατάλληλες άδειες από το χρήστη μας δίνει πρόσβαση στον/στους αισθητήρες που χρειαζόμαστε και η δεύτερη μας μεταφέρει τα δεδομένα που ζητάμε από τους αισθητήρες.

Η τάξη «SensorManager» περιλαμβάνεται στο πακέτο «android.hardware.SensorManager» και η τάξη «SensorEventListener» στο πακέτο «android.hardware.SensorEventListener».

##### 3.2.3 Βιβλιοθήκες εφαρμογής

Στο Gradle Script της εφαρμογής πρέπει να οριστούν οι βιβλιοθήκες που απαιτούνται για τη λειτουργία της.

Στην παρούσα εφαρμογή οι βιβλιοθήκες που απαιτούνται είναι οι παρακάτω:

- 'com.android.support:appcompat-v7:26.1.0'
- 'com.android.support.constraint:constraint-layout:1.0.2'
- 'jtds-1.2.8.jar'

Η `appcompat-v7` είναι η βιβλιοθήκη που μας προσφέρει τα απαραίτητα πακέτα για την υποστήριξη παλιότερων εκδόσεων Android APIs, καθώς επίσης και βασικές μεθόδους και τάξεις για την υλοποίηση της εφαρμογής.

Η `constraint-layout` είναι η βιβλιοθήκη που μας επιτρέπει το σχεδιασμό δυναμικών UI (περιβάλλον χρήστη) στο Android.

Οι δύο παραπάνω βιβλιοθήκες βρίσκονται μέσα στην «`com.android.support`».

Η `jtids-1.2.8.jar` είναι βιβλιοθήκη επικοινωνίας client – server και χρησιμοποιείται για την αποστολή των δεδομένων από την κινητή συσκευή στο Microsoft Azure SQL Server.

### 3.2.4 Αρχιτεκτονική android application



#### Διάγραμμα 1.UML τάξεων

Όπως φαίνεται στο Διάγραμμα 1 το android app αποτελείται από 9 τάξεις.

- ❖ Από αυτές οι `Fall` και `User` έχουν βοηθητικό ρόλο και υπάρχουν ώστε να μπορούμε να φτιάξουμε αντικείμενα τα οποία έχουν ως attributes τις πληροφορίες που θέλουμε να αποθηκεύσουμε στη βάση δεδομένων.

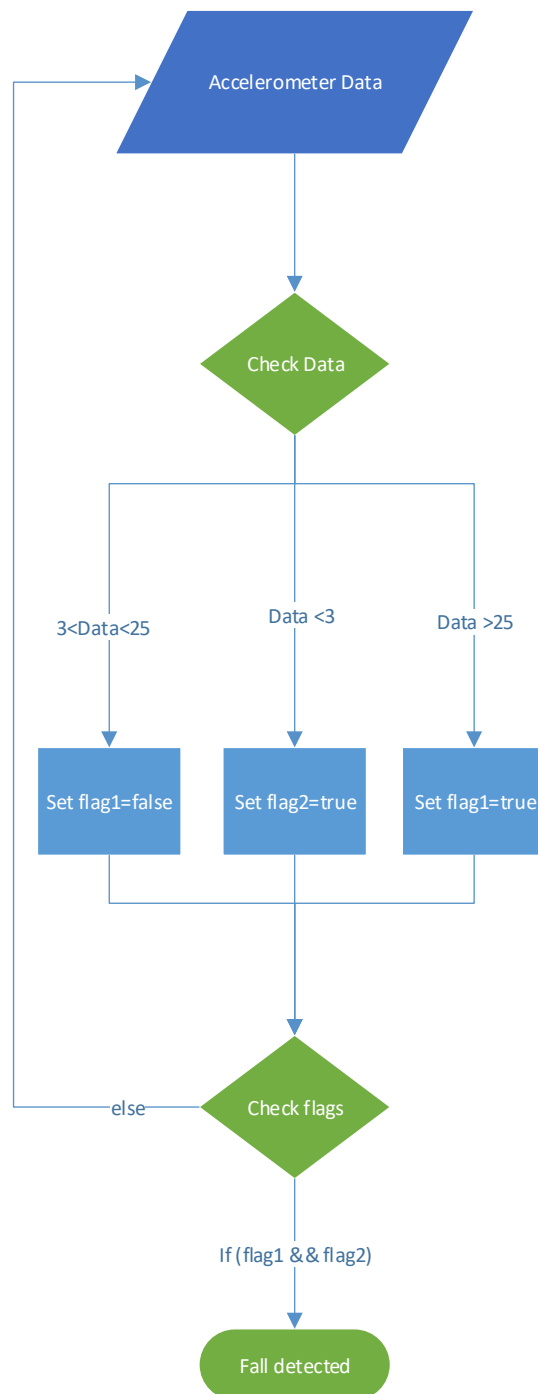
- ❖ Η τάξη `GetLocation` είναι υπεύθυνη για την παροχή των συντεταγμένων της συσκευής. Για να το καταφέρει αυτό, κάνουμε `implement` έναν `LocationListener` και μέσω της δημιουργίας ενός αντικειμένου από την τάξη `LocationManager` αποκτάμε πρόσβαση στα δεδομένα που μπορεί να μας παρέχει η συσκευή.
- ❖ Η `Dbhelper` είναι η τάξη που είναι υπεύθυνη για την σύνδεση με τη βάση δεδομένων και για την αποστολή δεδομένων σε αυτή. Για τη σύνδεση χρησιμοποιούμε ένα αντικείμενο από την τάξη `Connection` και αυτή αρχικοποιείται με την παρακάτω μέθοδο.

```
public Connection connectionclass(String database)
{
    StrictMode.ThreadPolicy policy = new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    Connection connection = null;

    try
    {
        Class.forName("net.sourceforge.jtds.jdbc.Driver");
        //ConnectionURL = ;
        connection = DriverManager.getConnection(database);
    }
    catch (SQLException se)
    {
        Log.e("error here 1 : ", se.getMessage());
    }
    catch (ClassNotFoundException e)
    {
        Log.e("error here 2 : ", e.getMessage());
    }
    catch (Exception e)
    {
        Log.e("error here 3 : ", e.getMessage());
    }
    return connection;
}
```

Εικόνα 4.

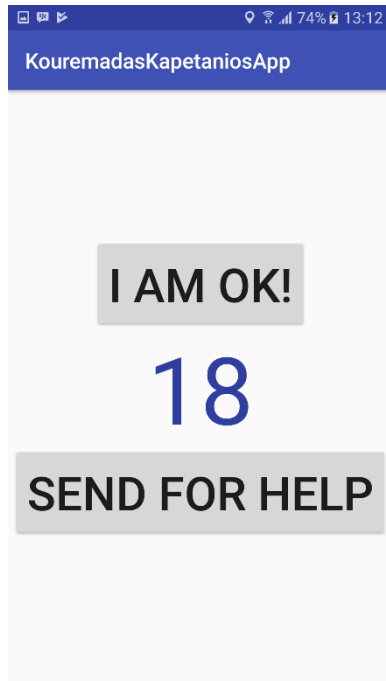
- ❖ Στην τάξη `Detector` υπάρχει η βασική λειτουργία της εφαρμογής μας, δηλαδή ο εντοπισμός της πτώσης. Η `Detector` κάνει `extend` την τάξη `service` και `implement` τη `SensorEventListener`. Η `service` μας επιτρέπει να έχουμε ένα αντικείμενο τύπου `Detector` το οποίο λειτουργεί συνέχεια (ακόμη και όταν η οθόνη της συσκευής είναι κλειστή και/ή κλειδωμένη). Για να επιτύχουμε τη συνεχόμενη λειτουργία στην μέθοδο της `Service` `onStartCommand`, επιστρέφουμε «`return START_STICKY;`». Αυτό δίνει σήμα στο λειτουργικό, ώστε το `service` μας, ακόμα και να καταστραφεί από τον `garbage collector`, να ξεκινήσει πάλι μόλις βρεθεί αρκετή `ram`.



**Διάγραμμα 2. Διάγραμμα διαδικασίας αναγνώρισης πτώσης**

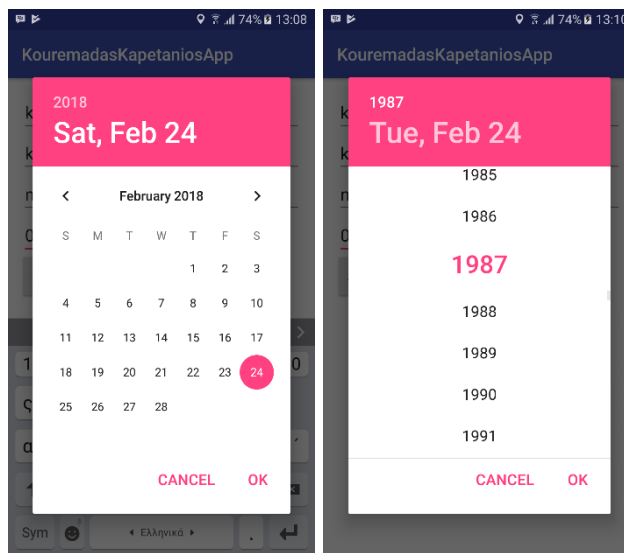
Στο διάγραμμα 2 φαίνεται ο αλγόριθμος που χρησιμοποιείται για τον εντοπισμό των πτώσεων. Τα δεδομένα του επιταχυνσιόμετρου (Accelerometer Data) τα παίρνουμε μέσω ενός `SensorManager` και μέσα στη μέθοδο `onSensorChanged(SensorEvent event)`, η οποία μας προσφέρεται από την τάξη `SensorEventListener` που κάναμε implement, τρέχουμε τον αλγόριθμό μας.

- ❖ Η τάξη AlarmWindow σε συνδυασμό με το activity\_alarm\_window.xml δημιουργούν ένα activity, το οποίο με κατάλληλα flags έχει την ιδιότητα να «ξυπνάει» τη συσκευή και να εμφανίζεται στο προσκήνιο (ακόμα και όταν η συσκευή είναι κλειδωμένη).



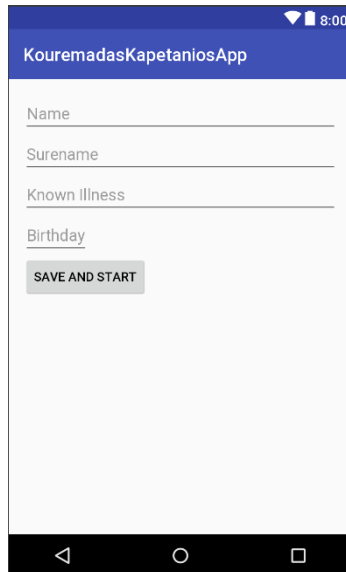
Εικόνα 5.

- ❖ Η τάξη setDate δημιουργεί το ημερολόγιο που παρέχουμε στο χρήστη, ώστε να μας δώσει την ημερομηνία γέννησης του. Κάνει implement τις View.OnFocusChangeListener και DatePickerDialog.OnDateSetListener, ώστε να μπορεί να δημιουργεί το οπτικό μέρος του ημερολογίου και ταυτόχρονα να μπορούμε να πάρουμε την τιμή που επιλέγει ο χρήστης.



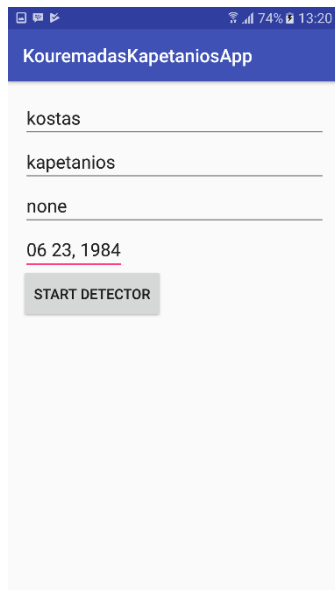
Εικόνα 6.

- ❖ Η PanikButtonWidget μέσω του « extends AppWidgetProvider » δημιουργεί ένα widget, του οποίου η λειτουργία είναι στο πάτημά του να στέλνει στο server έκκληση για βοήθεια. Επειδή η λειτουργικότητα που προσφέρουν τα widgets είναι περιορισμένη, έπρεπε αναγκαστικά να γράψουμε και εδώ τον ίδιο κώδικα με αυτόν που γράψαμε και στην GetLocation τάξη.
- ❖ Τέλος η MainActivity είναι η τάξη που δένει όλες τις παραπάνω μεταξύ τους. Είναι η τάξη από την οποία ξεκινάει το πρόγραμμα και δίνει στο χρήστη μέσω του activity\_main.xml την δυνατότητα να εγγραφεί στο σύστημα και να ξεκινήσει το service μας.



**Εικόνα 7.Οθόνη νέου χρήστη**

Εάν ο χρήστης έχει ήδη εγγραφεί στην εφαρμογή, τότε η εικόνα που βλέπει ο χρήστης είναι διαφορετική.



**Εικόνα 8. Οθόνη χρήστη που έχει ήδη εγγραφεί στην εφαρμογή**



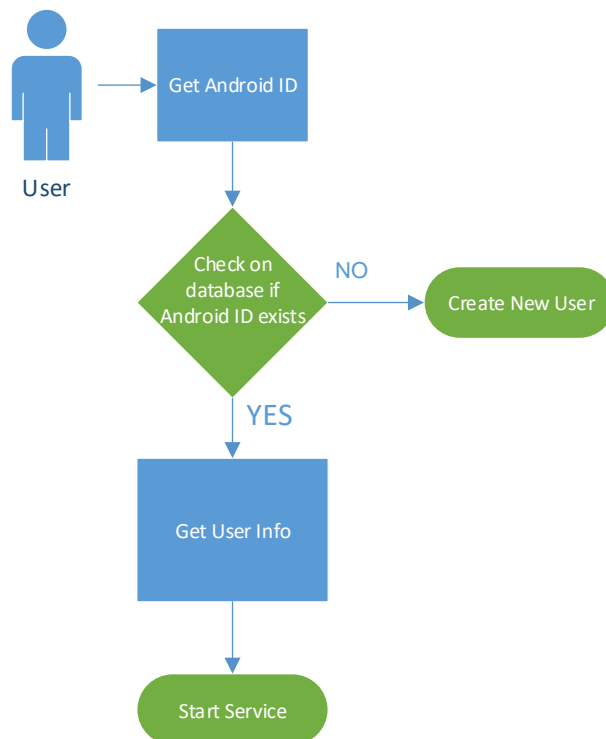
### 3.3 Διαγράμματα περιπτώσεων χρήσης

Στο σύστημά μας υπάρχουν οι εξής συμμετέχοντες:

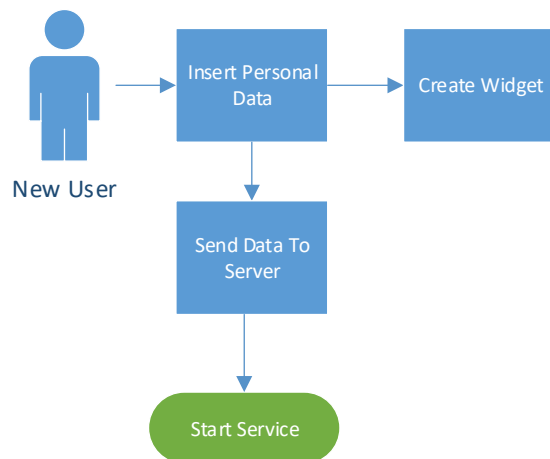
- Ο χρήστης (User)
- Ο χειριστής (Operator)

Ως χρήστης ορίζεται το άτομο που έχει στην κατοχή του την κινητή συσκευή στην οποία υπάρχει το android app.

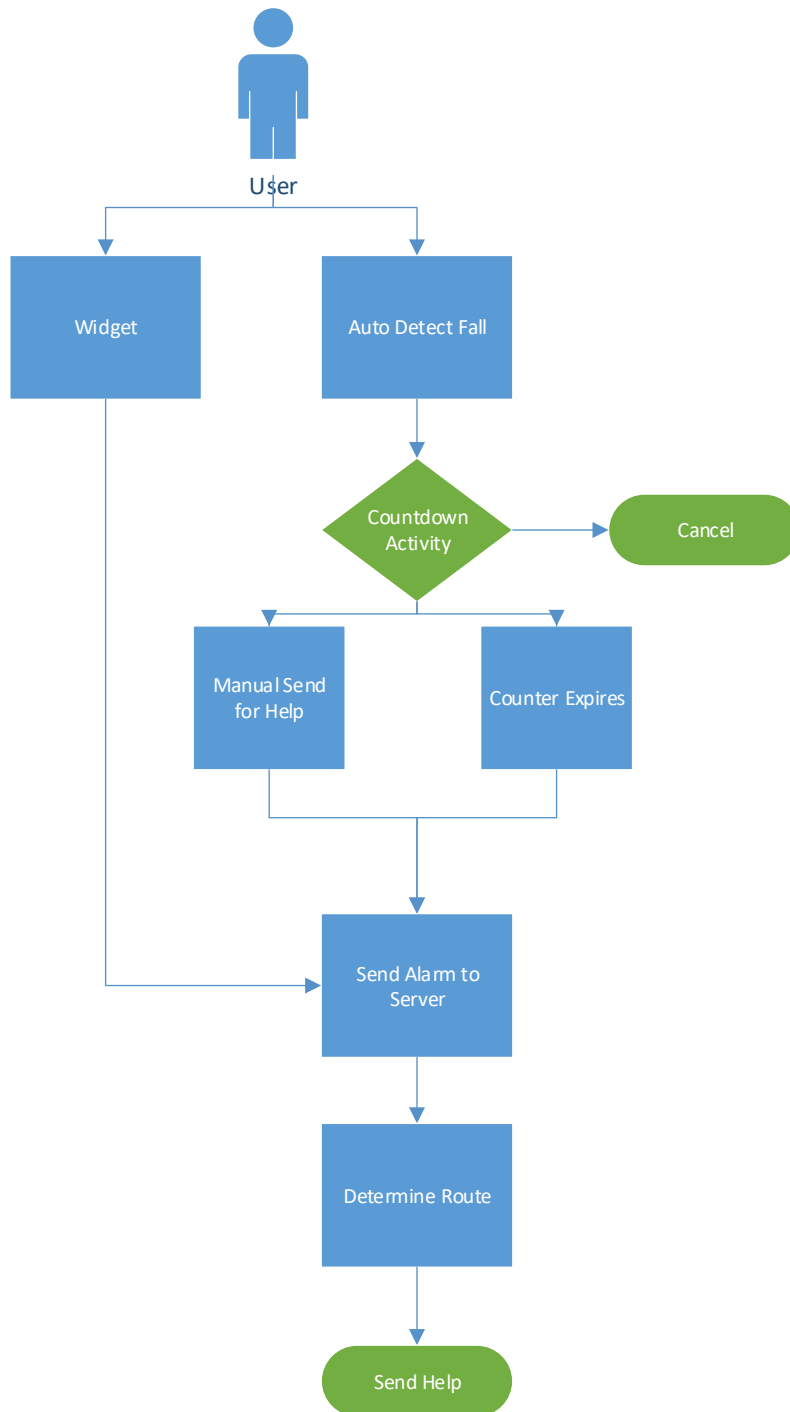
Ως χειριστής ορίζεται το άτομο που εποπτεύει το web app και είναι υπεύθυνο για την αποστολή βοήθειας. Ο χειριστής μπορεί να είναι είτε κάποιος υπάλληλος σε κάποιο σταθμό βοήθειας, είτε οδηγός ασθενοφόρου, κα.



Διάγραμμα 3. Φάση εκκίνησης Android app



**Διάγραμμα 4. Διάγραμμα δημιουργίας νέου χρήστη**

**Διάγραμμα 5.**

Στο Διάγραμμα 5 αποτυπώνονται οι εξής λειτουργίες:

- Η χρήση του Widget
- Η διαδικασία που ακολουθείται, όταν αναγνωρισθεί πτώση

Με τη χρήση του Widget ο χρήστης αποστέλλει άμεσα αίτημα βοήθειας.

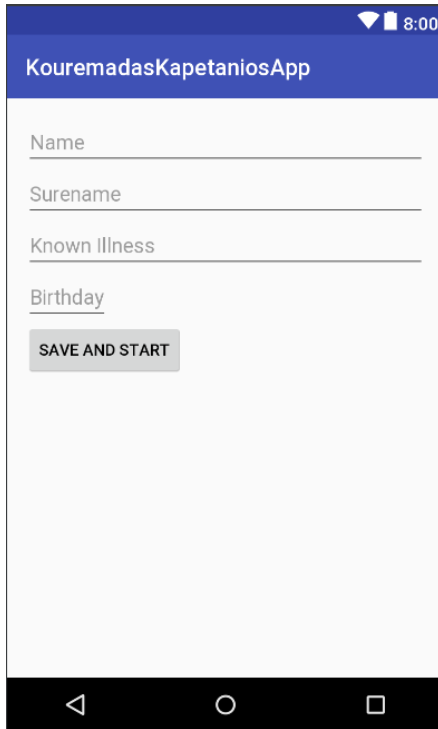
Όταν εντοπισθεί πτώση, το σύστημα ενημερώνει το χρήστη με ήχο και εικόνα πως έχει εντοπίσει πτώση. Ο χρήστης έχει στη διάθεσή του 30 δευτερόλεπτα για να ακυρώσει το αίτημα βοήθειας. Εάν παρέλθει ο χρόνος το σύστημα αποστέλλει αυτόματα αίτημα βοήθειας. Επιπλέον, παρέχεται η δυνατότητα να παρακαμφθεί ο χρονοδιακόπτης από το χρήστη, και να αποσταλεί άμεσα αίτημα βοήθειας.

## 4. Εγχειρίδιο Χρήστη

### 4.1 Σύντομη παρουσίαση Android Application

Κατά την εκκίνηση της εφαρμογής μέσω παρασκηνακού ελέγχου διαπιστώνεται αν ο χρήστης είναι εγγεγραμμένος ή όχι.

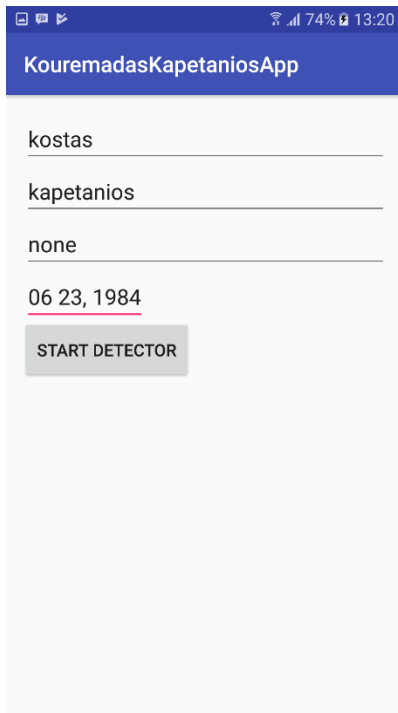
Εάν δεν είναι εγγεγραμμένος εμφανίζεται η ακόλουθη φόρμα.



Εικόνα 9. Εγγραφή νέου χρήστη

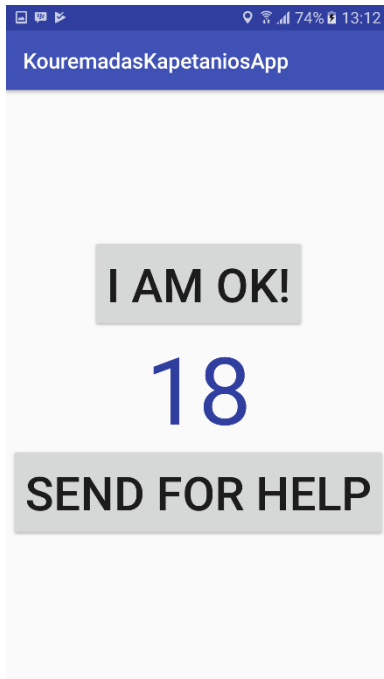
Θα πρέπει να συμπληρώσει τα στοιχεία του και να πατήσει το κουμπί «SAVE AND START». Αυτή η ενέργεια εγγράφει το χρήστη στη βάση δεδομένων και η περαιτέρω ταυτοποίησή του γίνεται μέσω του Android ID.

Σε περίπτωση που ο χρήστης κάνει επανεκκίνηση τη συσκευή του, αρκεί να εκκινήσει εκ νέου την εφαρμογή, η οποία μέσω του Android ID θα βρει τα στοιχεία του χρήστη και θα του εμφανίσει την ακόλουθη φόρμα με τα στοιχεία του συμπληρωμένα.



**Εικόνα 10.**

Όταν το app αναγνωρίσει κάποια πτώση, έρχεται στο προσκήνιο η φόρμα της εικόνας 11, μέσω της οποίας ο χρήστης μπορεί να ακυρώσει, είτε να επισπεύσει την έκκληση βοήθειας. Η επίσπευση γίνεται μέσω του κουμπιού «SEND FOR HELP», ενώ η ακύρωση μέσω του «I AM OK!». Εάν ο χρήστης δεν είναι σε θέση να χρησιμοποιήσει τη συσκευή του, τότε ο χρονοδιακόπτης των 30 δευτερολέπτων θα αποστείλει αυτόματα αίτημα για βοήθεια.



**Εικόνα 11.**

Δεδομένης της ανάγκης να μπορεί ο χρήστης να καλέσει σε βοήθεια πριν εκδηλωθεί το περιστατικό (πχ σε περιπτώσεις εμφράγματος), δημιουργήθηκε το Widget «EMERGENCY BUTTON», το οποίο φαίνεται στην εικόνα12. Ο χρήστης έχει τη δυνατότητα να το τοποθετήσει σε όποιο σημείο της επιφάνειας εργασίας του επιθυμεί. Με το πάτημα του συγκεκριμένου κουμπιού στέλνεται αμέσως αίτημα για βοήθεια, με τις συντεταγμένες και τα προσωπικά στοιχεία του χρήστη.



**Εικόνα 12.**

## 5. Συμπεράσματα και μελλοντικές επεκτάσεις

Στην παρούσα μεταπτυχιακή αναπτύχθηκε και παρουσιάστηκε το frontend μιας fullstack εφαρμογής η οποία ανιχνεύει και συλλέγει δεδομένα πτώσεων και τα απεικονίζει με κατάλληλο τρόπο.

Το πρόβλημα το οποίο εντοπίσαμε και προσπαθήσαμε να επιλύσουμε με την εφαρμογή μας είναι η εξάλειψη του ενδιάμεσου και η άμεση ειδοποίηση του συστήματος υγείας. Αυτό θα συμβάλει στη βελτίωση των παροχών υγείας και στη μείωση των σοβαρών συνεπειών από τις πτώσεις.

Τα δεδομένα που συλλέγονται αφορούν την τοποθεσία του χρήστη και προσωπικά του δεδομένα, όπως όνομα, επώνυμο, ηλικία και πιθανές ασθένειες.

Το σύστημά μας επιδέχεται αρκετές επεκτάσεις και βελτιώσεις, για να φτάσει σε επίπεδο εμπορικής εφαρμογής.

Ενδεικτικά προτάσσονται τα παρακάτω:

- Προσθήκη ρυθμίσεων, όπως ευαισθησία ανίχνευσης πτώσης, αλλαγή ήχου ειδοποίησης πτώσης (και στο χρήστη αλλά και στο χειριστή)
- Παροχή οδηγιών χρήσης εντός της εφαρμογής
- Απεικόνιση ιστορικού προσωπικών κλήσεων βοήθειας εντός του android app
- Βελτίωση του αλγόριθμου ή επιλογή άλλου αλγόριθμου.
- Σύνδεση με wearable και εντοπισμός εκτάκτων περιστατικών μέσω των αισθητήρων του wearable.
- Μελέτη και εφαρμογή για συμμόρφωση με το GDPR. Το GDPR είναι ο ευρωπαϊκός κανονισμός για την προστασία προσωπικών δεδομένων ο οποίος τίθεται σε ισχύ από τις 25 Μαΐου 2018.
- Έλεγχος ύπαρξης σύνδεσης στο διαδίκτυο και ενημέρωση του χρήστη, εάν δεν υπάρχει.



## **6. Βιβλιογραφία**

<https://stackoverflow.com/>

<https://msdn.microsoft.com/en-us/dn308572.aspx>

<https://developer.android.com/index.html>

<https://docs.microsoft.com/en-us/azure/>

[https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#wf\\_mvc\\_wp](https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx#wf_mvc_wp)

<https://azure.microsoft.com/en-us/offers/ms-azr-0144p/>

<https://azure.microsoft.com/en-us/free/free-account-faq/>

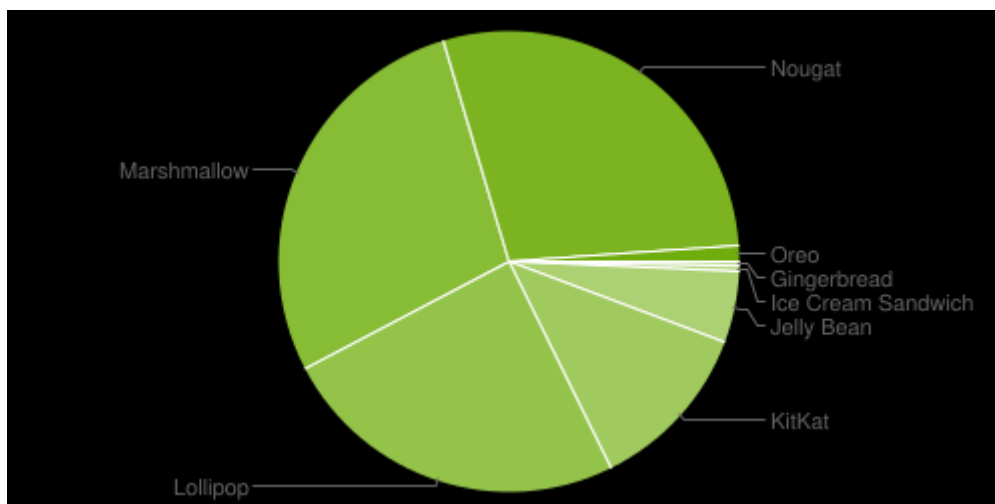
<https://www.eugdpr.org/>

<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>

Ανδρέας Κουρεμάδας {Σχεδίαση και ανάπτυξη διαδικτυακής εφαρμογής ενημέρωσης και διαχείρισης πτώσεων}

## 7. Παράρτημα

### 7.1 Κατανομή αγοράς λειτουργικών Android



Εικόνα 12.

Version	Codename	API	Distribution
<a href="#">2.3.3 -</a>	Gingerbread	10	0.30%
<a href="#">2.3.7</a>			
<a href="#">4.0.3 -</a>	Ice Cream Sandwich	15	0.40%
<a href="#">4.0.4</a>			
<a href="#">4.1.x</a>	Jelly Bean	16	1.70%
<a href="#">4.2.x</a>		17	2.60%
<a href="#">4.3</a>		18	0.70%
<a href="#">4.4</a>	KitKat	19	12.00%
<a href="#">5</a>	Lollipop	21	5.40%
<a href="#">5.1</a>		22	19.20%
<a href="#">6</a>	Marshmallow	23	28.10%
<a href="#">7</a>	Nougat	24	22.30%
<a href="#">7.1</a>		25	6.20%
<a href="#">8</a>	Oreo	26	0.80%
<a href="#">8.1</a>		27	0.30%

Data collected during a 7-day period ending on February 5, 2018.

Any versions with less than 0.1% distribution are not shown.

<https://developer.android.com/about/dashboards/index.html>

## 7.2 Κώδικας αναγνώρισης πτώσεων

```
Boolean flag1=false;
Boolean flag2=false;

@Override
public void onSensorChanged(SensorEvent event) {
    if(acc.getType()==Sensor.TYPE_ACCELEROMETER){

        double x=event.values[0];
        double y=event.values[1];
        double z=event.values[2];
        Double metro=Math.sqrt(Math.pow(x,2)+Math.pow(y,2)+Math.pow(z,2));

        if(metro>=12){
            flag1=true;
        }
        if(metro<3){
            flag2=true;
        }
        if(metro>=3&&metro<15){
            flag1=false;
            flag2=false;
        }
        if(flag1&&flag2){
            locD.findMe();

            Toast.makeText(getApplicationContext(),"fall
detected",Toast.LENGTH_LONG).show();
            Intent dialogIntent = new Intent(this, AlarmWindow.class);
            dialogIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(dialogIntent);
            flag1=false;
            flag2=false;
        }
    }
}
```

Εικόνα 13.

### 7.3 Κώδικας χρονοδιακόπτη

```

new CountdownTimer(30000, 1000) {

    @Override
    public void onTick(long millisUntilFinished) {
        Long timeToShow=millisUntilFinished/1000;
        timer.setText(timeToShow.toString());
        if(flag) {
            Toast.makeText(getApplicationContext(),"Sending
data",Toast.LENGTH_SHORT).show();
            this.cancel();
            sendData(getApplicationContext(),true);

            finish();
        }
        if(flag2){
            Toast.makeText(getApplicationContext(),"I'm huppy you are
OK!",Toast.LENGTH_SHORT).show();
            this.cancel();
            sendData(getApplicationContext(),false);
            finish();
        }
    }

    @Override
    public void onFinish() {
        timer.setText("0");
        sendData(getApplicationContext(),true);
        Toast.makeText(getApplicationContext(),"Sending
data",Toast.LENGTH_SHORT).show();
    }

}.start();

```

Εικόνα 14.

### 7.4 Permissions Εφαρμογής και λοιπά attributes στο Manifest

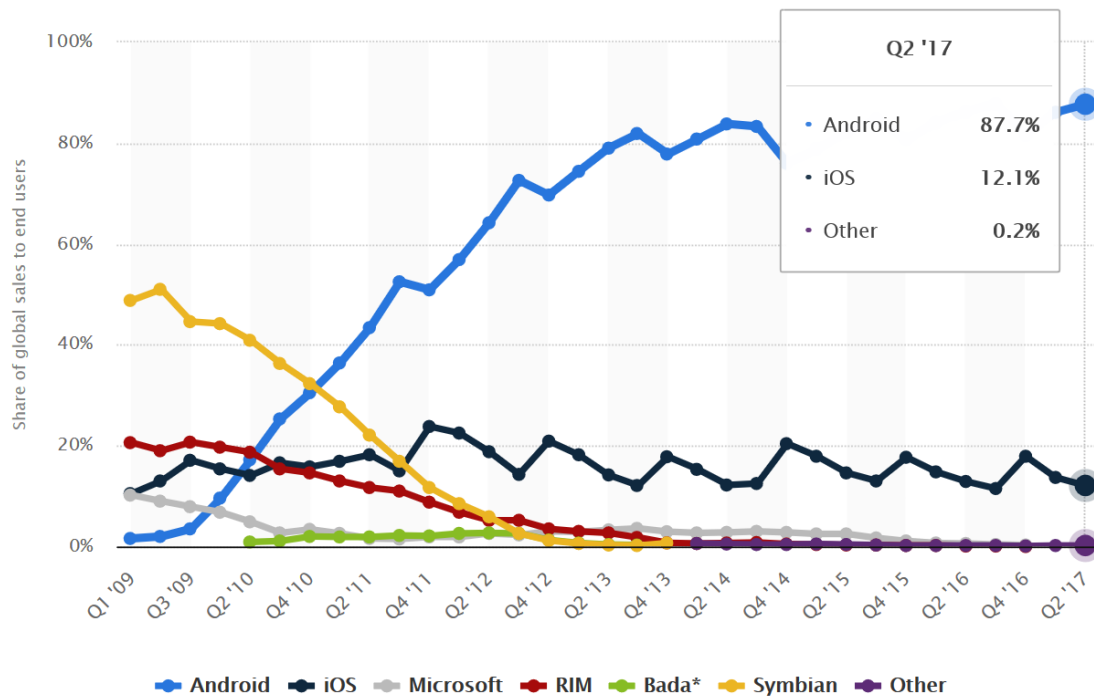
Για τη λειτουργία της εφαρμογής απαιτούνται τα παρακάτω permissions

- android.permission.INTERNET
- android.permission.ACCESS\_COARSE\_LOCATION
- android.permission.ACCESS\_FINE\_LOCATION
- android.permission.WAKE\_LOCK

Επίσης στο Manifest της εφαρμογής στο κομμάτι που αφορά το Activity που ενεργοποιεί ο εντοπισμός πτώσης χρειάζεται το παρακάτω attribute, ώστε να μπορεί να φαίνεται πάνω από lockScreen.

```
android:showOnLockScreen="true"
```

### 7.5 Μεριδίο της αγοράς (λειτουργικά συστήματα)



Εικόνα 15. <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>