



ANDROID APPLICATION

**SAVE THE TEACHER**

ΜΕΤΑΠΤΥΧΙΑΚΟΣ ΦΟΙΤΗΤΗΣ

**ΣΠΥΡΙΔΩΝ ΚΛΕΦΤΑΚΗΣ**  
**(ΜΕ1621)**

---

# SAVE THE TEACHER

*Εφαρμογή σε Android*

---

Πανεπιστήμιο Πειραιά  
Τμήμα Ψηφιακών Συστημάτων



**Εφαρμογή σε Android: “Save The Teacher”**

Κλεφτάκης Σπυρίδων (ΜΕ 1621)

Μάθημα: Κινητή Υπολογιστική και Εφαρμογές

Διδάσκων Καθηγητής: Μενύχτας Ανδρέας

ΗΜΕΡΟΜΗΝΙΑ

---

Ιουν 2017



## Περιεχόμενα

1. Εισαγωγή .....	4
2. Γενική Ιδέα Εφαρμογής .....	4
3. Τεχνολογίες Υλοποίησης .....	4
3.1. VM στον ~Okeanos ( <a href="https://okeanos.grnet.gr/home/">https://okeanos.grnet.gr/home/</a> ) .....	5
3.2. MYSQL (Βάση Δεδομένων) .....	6
3.3. PHP .....	8
3.3.1. DBinfo.php .....	8
3.3.2. IsFollowing.php .....	8
3.3.3. Login.php .....	9
3.3.4. Register.php .....	9
3.3.5. TweetAdd.php .....	10
3.3.6. TweetList.php .....	10
3.3.7. UserFollowing.php .....	10
3.4. Firebase Login .....	11
3.5. Google Sign In .....	12
3.6. Firebase Notifications .....	15
3.7. Realm .....	16
3.8. Picasso .....	17
3.9. Firebase Storage .....	17
3.10. Retrofit API .....	18
3.11. Firebase Crash Reporting .....	19
4. Σύντομος Οδηγός Χρήστη .....	20
4.1. Είσοδος στην εφαρμογή .....	20
4.2. Κύρια Οθόνη Εφαρμογής .....	22
4.3. Καιρός στον Κόσμο .....	26
4.4. Πρόγνωση Καιρού .....	26
4.5. Περιοχές Διευθύνσεων Πρωτοβάθμιας Εκπαίδευσης .....	27
4.6. Λοιπά Χαρακτηριστικά .....	28
5. Συμπεράσματα – Προβλήματα – Μελλοντική Εργασία .....	28





## 1. Εισαγωγή

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια της απαλλακτικής εργασίας του Β' Εξαμήνου στο μάθημα Κινητή Υπολογιστική και Εφαρμογές με θέμα: «Ανάπτυξη Εφαρμογής σε Android».

## 2. Γενική Ιδέα Εφαρμογής

Η εφαρμογή αποτελεί μία κοινωνική πλατφόρμα στοχευμένη για τους αναπληρωτές δασκάλους ανά την Ελλάδα. Η γενική ιδέα είναι ότι ο κάθε δάσκαλος (που κάθε χρόνο βρίσκεται σε διαφορετικό σχολείο - αναπληρωτής) κάνοντας ένα login στην εφαρμογή μπορεί να δημοσιεύει το μέρος στο οποίο τον πήρανε για εργασία ή να κοινοποιεί διάφορους προβληματισμούς του, ώστε να βρει κι άλλους δασκάλους προκειμένου να “κοινωνικοποιηθεί” στην νέα του περιοχή (για παράδειγμα να κανονίσει να πηγαίνουν με κοινό αυτοκίνητο στο σχολείο). Επιπροσθέτως, μπορεί να δημοσιεύει διάφορες σκέψεις προς όσους τον ακολουθούν και να βλέπει «σκέψεις» αυτών που ακολουθεί ή να κάνει αναζητήσεις με λέξεις κλειδιά που τον ενδιαφέρουν. Επίσης παρέχεται η δυνατότητα πρόγνωσης του καιρού μέσα από την εφαρμογή. Τέλος υπάρχει η δυνατότητα ο δάσκαλος να αναζητήσει όλες τις περιοχές, τις διευθύνσεις και τα σχολεία σε κάθε πρωτοβάθμια διεύθυνση της Ελλάδος.

## 3. Τεχνολογίες Υλοποίησης

Στην εφαρμογή χρησιμοποιήθηκε ένας μεγάλος αριθμός τεχνολογιών, οι οποίες διδάχθηκαν εντός των εργαστηρίων κατά την διάρκεια των μαθημάτων του Β' εξαμήνου στο συγκεκριμένο μάθημα. Η εφαρμογή υλοποιήθηκε με Android Studio.

Οι τεχνολογίες επιγραμματικά είναι οι κάτωθι:

- Virtual Machine (VM) : Δημιουργήθηκε για τις ανάγκες της εφαρμογής ένα VM στον ~oceanos (<https://oceanos.grnet.gr/home/>), όπου στήθηκε ένας Microsoft Windows Server 2012, εγκαταστάθηκε το WAMP και



ρυθμίστηκε η κατάλληλη πόρτα (localhost:8083) καθώς και τα κατάλληλα configuration files του apache, όπως και το firewall του server.

- MySQL: Φτιάχτηκε βάση δεδομένων(ΒΔ) σε SQL στον παραπάνω server
- PHP: Υλοποιήθηκαν επτά (7) scripts σε php αρχεία για την διαχείριση της ΒΔ.
- Firebase Login: Χρησιμοποιήθηκε το firebase για Login στην εφαρμογή
- Google Login: Χρησιμοποιήθηκε το Google Sign In για εναλλακτικό Login των χρηστών.
- Firebase Notifications: Χρησιμοποιήθηκε η τεχνολογία Firebase Cloud Messaging για να στέλνονται notifications στην εφαρμογή
- Realm: Χρησιμοποιήθηκε το Realm για διαχείριση ενός JSON αρχείου με τις περιοχές των αναπληρωτών και δημιουργία από αυτό ενός Realm Object και στη συνέχεια κατάλληλης λίστας με τις περιοχές.
- Picasso: Χρησιμοποίηση του Picasso για εισαγωγή φωτογραφιών στη δημοσίευση σκέψεων.
- Firebase Storage: Αποθήκευση φωτογραφιών των χρηστών στο firebase storage συγκεκριμένα στο "gs://savetheteacherapp-55d47.appspot.com"
- Retrofit API: Έχει υλοποιηθεί με retrofit api του <http://openweathermap.org/> για πρόγνωση καιρού μέσα από την εφαρμογή τόσο για τον καιρό που επικρατεί όσο και για πρόγνωση επτά ημερών. Επίσης έχει υλοποιηθεί με άλλο api η αποτύπωση καιρού και συντεταγμένων οποιασδήποτε περιοχής ερωτήσει ο χρήστης.
- Firebase Crash Reporting: Χρησιμοποίηση για έλεγχο των σφαλμάτων της εφαρμογής που έχει εγκατασταθεί στους users.

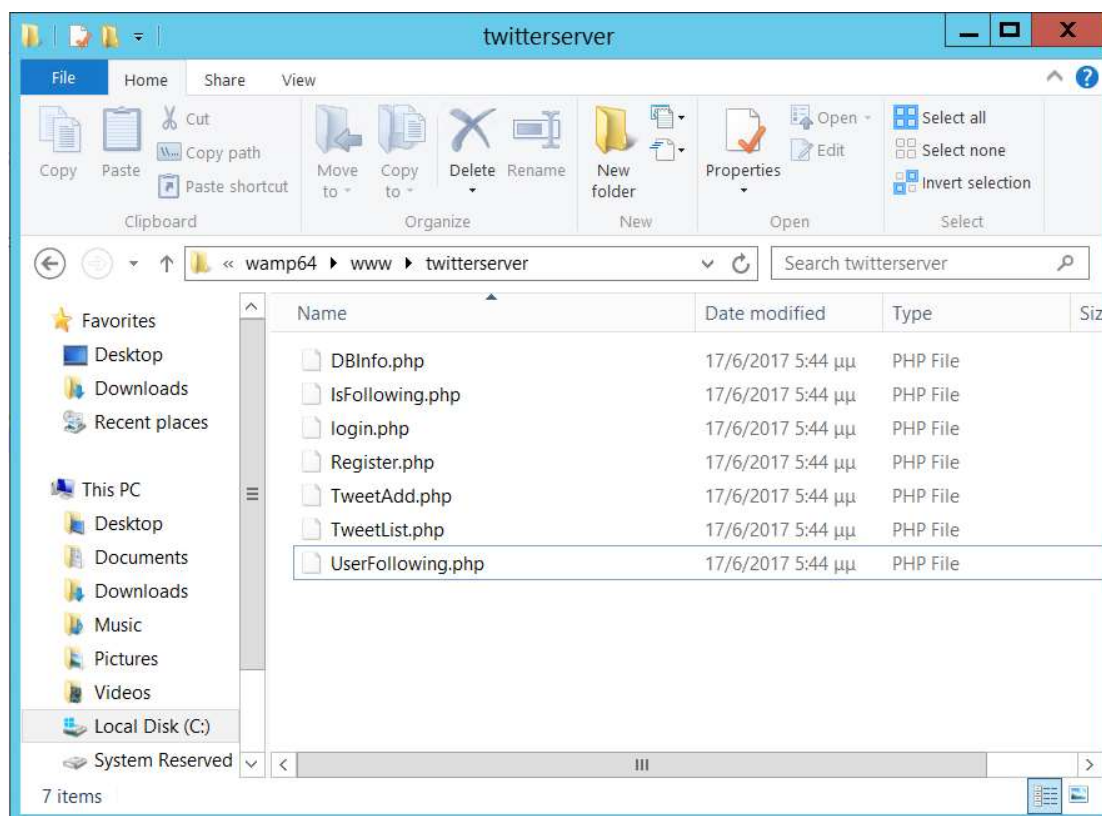
Αναλυτικά για το καθένα από τα παραπάνω:

### 3.1. VM στον ~Okeanos (<https://okeanos.grnet.gr/home/>)

Ο Server και η Βάση Δεδομένων της εφαρμογής αναπτύχθηκαν σε ένα VM που δημιουργήθηκε στον ~okeanos με λειτουργικό Windows Server 2012. Σε αυτόν

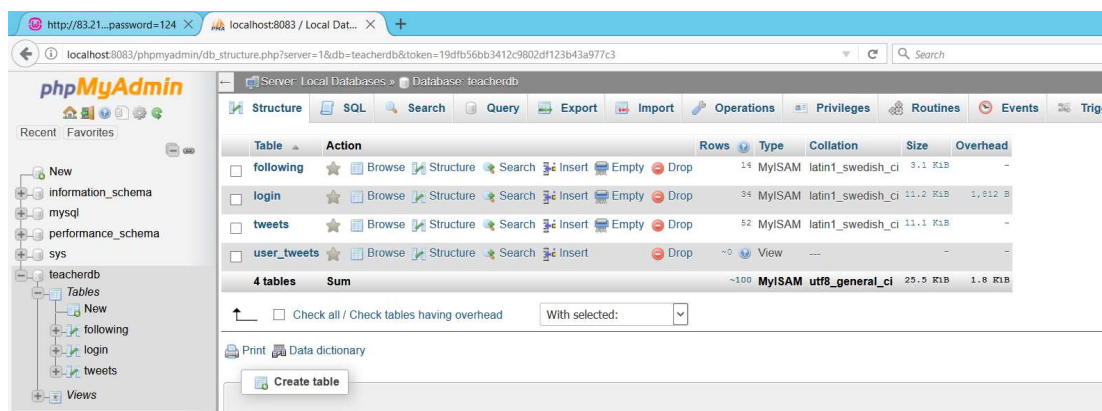


ρυθμίστηκε κατάλληλα το firewall, εγκαταστάθηκε ο Wamp Server, προκειμένου να εγκατασταθεί και ο apache server, ρυθμίστηκαν κατάλληλα τα αρχεία configurations για να μπορεί κάποιος να «χτυπάει» τον server. Εν συνεχεία στον φάκελο wamp64/www/ τοποθετήθηκαν τα αρχεία php που δημιουργήθηκαν για επικοινωνία με τη ΒΔ:



### 3.2. MYSQL (Βάση Δεδομένων)

Κατασκευάστηκε η ΒΔ με τη βοήθεια της MYSQL οπότε και προέκυψε η παρακάτω ΒΔ teacherdb με τους πίνακες που θα χρειαστούμε στην εφαρμογή:







Αναλυτικότερα με τα πεδία τους υπάρχουν οι παρακάτω πίνακες:

teacherdb tweets	
🔑	tweet_id : int(11)
#	user_id : int(11)
📄	tweet_text : varchar(100)
📄	tweet_picture : varchar(350)
📅	tweet_date : timestamp

teacherdb user_tweets	
#	tweet_id : int(11)
📄	tweet_text : varchar(100)
📄	tweet_picture : varchar(350)
📅	tweet_date : timestamp
#	user_id : int(11)
📄	first_name : varchar(50)
📄	picture_path : varchar(350)

teacherdb following	
#	user_id : int(11)
#	following_user_id : int(11)

teacherdb login	
🔑	user_id : int(11)
📄	first_name : varchar(50)
📄	email : varchar(50)
📄	password : varchar(50)
📄	picture_path : varchar(350)

(δεν κρίνεται σκόπιμο να αναλυθούν περαιτέρω)

καθώς και ένα view για τα tweets («σκέψεις» στην εφαρμογή) όσων κάνουμε follow !

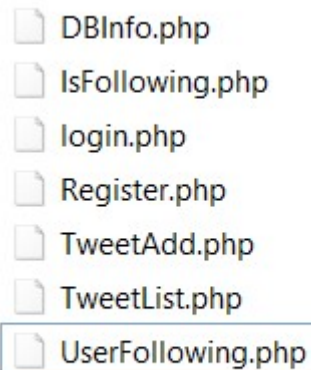
The screenshot shows the phpMyAdmin interface for the 'teacherdb' database. The left sidebar shows the database structure with 'teacherdb' expanded, showing tables 'following', 'login', 'tweets', and 'user\_tweets'. The main panel shows the 'user\_tweets' table structure with columns: 'tweet\_id' (int(11)), 'tweet\_text' (varchar(100)), 'tweet\_picture' (varchar(350)), and 'tweet\_date' (timestamp). The 'Create table' dialog is open, showing 'Name:' and 'Number of columns: 4'.





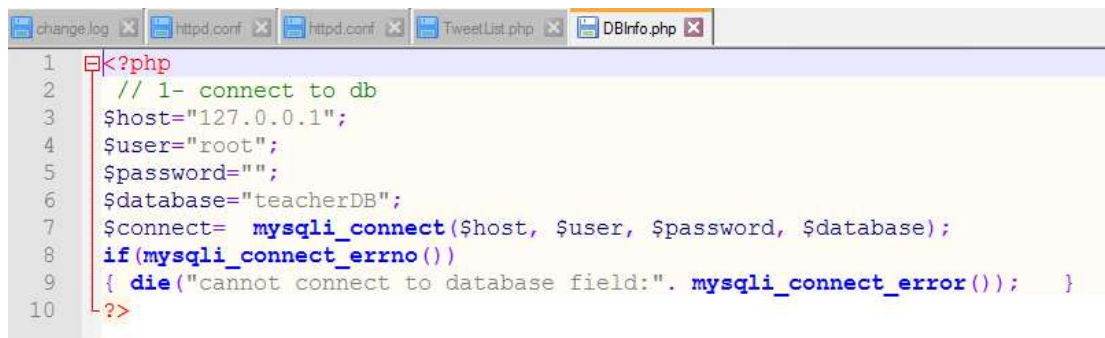
### 3.3. PHP

Στη συνέχεια κατασκευάστηκαν επτά αρχεία php γλώσσας προκειμένου να γίνεται η επικοινωνία μεταξύ του server και της εφαρμογής. Τα αρχεία που δημιουργήθηκαν φαίνονται αναλυτικά παρακάτω:



#### 3.3.1. DBinfo.php

Περιέχει της πληροφορίες για την ΒΔ. Τα υπόλοιπα αρχεία καλούν απλά το αρχείο αυτό αρχικά (require).



#### 3.3.2. IsFollowing.php

Ποιος user ακολουθεί ποιόν.



```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4
5 $query="select * from following where user_id=" . $_GET['user_id'].
6 " and following_user_id=" . $_GET['following_user_id'] ; // $username=$_GET['username'];
7
8 $result= mysqli_query($connect, $query);
9 if(!$result)
10 { die("Error in query");}
11 //get data from database
12 $output=array();
13 while($row= mysqli_fetch_assoc($result))
14 {
15     $output[]=$row; // $row['id']
16 }
17
18 if ($output) {
19     print("{'msg':'is subscriber'}");
20 }
21
22 else {
23     print("{'msg':'is not subscriber'}");
24 }
25
26 //5- close connection
27 mysqli_close($connect);
28 ?>
```

### 3.3.3. Login.php

Το login με όλα τα στοιχεία που χρειαζόμαστε προς έλεγχο

```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4
5 // define query
6 $query="select * from login where email= '$_GET['email']' and password= '$_GET['password']'"; // $username=$_GET['username'];
7 $result= mysqli_query($connect, $query);
8 if(!$result)
9 { die("Error in query");}
10 //get data from database
11 $output=array();
12 while($row= mysqli_fetch_assoc($result))
13 {
14     $output[]=$row; // $row['id']
15     break;
16 }
17 if ($output) {
18     print( "{'msg':'Pass Login' . ", 'info':". json_encode($output) . "}")// this will print the output in json
19 }
20
21 else{
22     print("{'msg':' cannot login'}");
23 }
24
25 // 4 clear
26 mysqli_free_result($result);
27 //5- close connection
28 mysqli_close($connect);
29 ?>
```

### 3.3.4. Register.php

Είναι η καταχώρηση στον φάκελο login όταν εγγράφεται αρχικά ο χρήστης

```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4 // define query
5 $query="insert into login(first_name,email,password,picture_path) values ('" . $_GET['first_name'] . "','" . $_GET['email'] . "','" . $_GET['password'] . "','" . $_GET['picture_path'] . "')";
6 $result= mysqli_query($connect, $query);
7 if(!$result)
8 { $output = "{'msg':'fail'}";
9 }
10
11 else {
12     $output = "{'msg':'user is added'}";
13 }
14
15 print( $output);// this will print the output in json
16
17 //5- close connection
18 mysqli_close($connect);
19 ?>
```



### 3.3.5. TweetAdd.php

Η προσθήκη στον πίνακα των tweets του tweet («σκέψης» στην εφαρμογή) που έγινε post.

```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4 // define query
5 //print($GET['tweet_picture']);
6 $query="insert into tweets(user_id,tweet_text,tweet_picture) values (" . $_GET['user_id'] . "," . $_GET['tweet_text'] . "," . $_GET['tweet_picture']
7 $result= mysqli_query($connect, $query);
8 if(!$result)
9 { $output = '{"msg':'fail'}";
10 }
11 else {
12 $output = '{"msg':'tweet is added'}";
13 }
14
15 print($output);// this will print the output in json
16
17 //5- close connection
18 mysqli_close($connect);
19 ?>
```

### 3.3.6. TweetList.php

Η λίστα με τα tweets προκειμένου να δημιουργήσουμε στον κώδικα αργότερα το search σε αυτό τον πίνακα.

```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4 // define query //StartFrom
5 if ($username=$_GET['op']==1) { // my following
6 $query="select * from user_tweets where user_id in (select following_user_id from following where user_id=" . $_GET['user_id'] . ")
7 or user_id=" . $_GET['user_id'] . " order by tweet_date DESC";
8 " LIMIT 20 OFFSET " . $_GET['StartFrom'] ; // $username=$_GET['username'];
9 }
10 elseif ($username=$_GET['op']==2) { // specific person post
11 $query="select * from user_tweets where user_id=" . $_GET['user_id'] . " order by tweet_date DESC";
12 " LIMIT 20 OFFSET " . $_GET['StartFrom'] ; // $username=$_GET['username'];
13 }
14 elseif ($username=$_GET['op']==3) { // search post
15 $query="select * from user_tweets where tweet_text like '%" . $_GET['query'] . "%'
16 " LIMIT 20 OFFSET " . $_GET['StartFrom'] ; // $username=$_GET['username'];
17 }
18
19 $result= mysqli_query($connect, $query);
20 if(!$result)
21 { die("Error in query");}
22 //get data from database
23 $output=array();
24 while($row= mysqli_fetch_assoc($result))
25 {
26 $output[]=$row; // $row['id']
27 }
28
29 if ($output) {
30 print( '{"msg':'has tweet'. ", 'info':". json_encode($output) ."}");// this will print the output in json
31 }
32 else{
33 print( '{"msg':'no tweet'}");
34 }
35
36 print(json_encode($output));// this will print the output in json
37
38 // 4 clear
39 mysqli_free_result($result);
40 //5- close connection
41 mysqli_close($connect);
42 ?>
```

### 3.3.7. UserFollowing.php

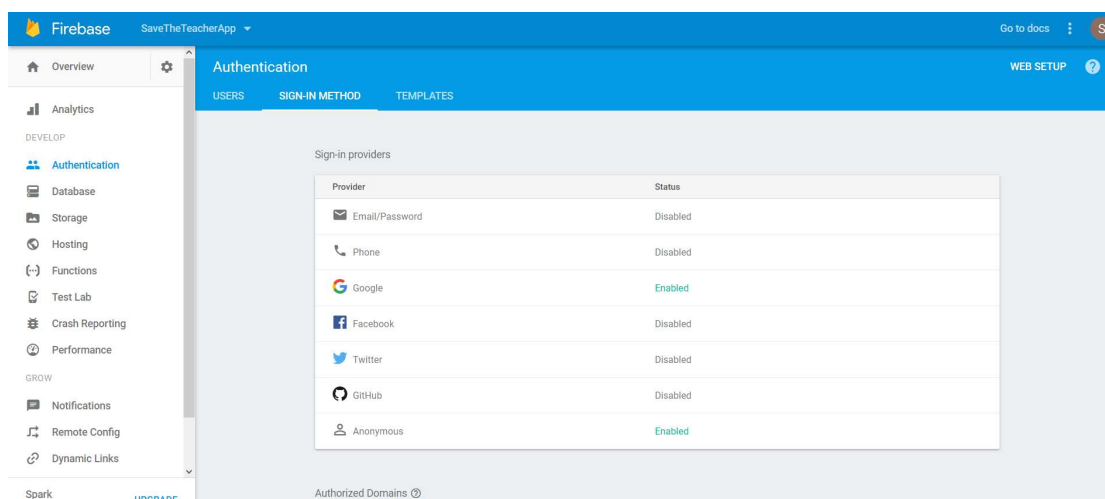
Είναι ο πίνακας που καταχωρούμε ποιος user ακολουθεί ποιόν.



```
1 <?php
2 // 1- connect to db
3 require("DBInfo.php");
4
5 // define query
6 if($_GET['op']==1) //op=1 add, op=2 delete
7 // add following
8 $query="insert into following(user_id,following_user_id) values (" . $_GET['user_id']. "," . $_GET['following_user_id'] . ")"; // $username=$_GET['use
9 }
10 else{ // remove following
11 $query="delete from following where user_id=" . $_GET['user_id']. " and following_user_id=" . $_GET['following_user_id'] ;
12 }
13 $result= mysqli_query($connect, $query);
14 if($result)
15 { $output =["msg":'fail'];
16 }
17 else {
18 $output =["msg":'following is updated'];
19 }
20
21 print( $output);// this will print the output in json
22
23 //5- close connection
24 mysqli_close($connect);
25
```

### 3.4. Firebase Login

Χρησιμοποιήθηκε το Firebase Login για την είσοδο στην εφαρμογή και συγκεκριμένα το anonymous login:



Ο κώδικας υλοποίησης της μεθόδου για το παραπάνω login βρίσκεται στη κλάση Login.java του κώδικα της εφαρμογής.



Συγκεκριμένα ο χρήστης μπορεί να ορίσει τα username, email και password για την είσοδο στην εφαρμογή.

### 3.5. Google Sign In

Εναλλακτικά για την είσοδο στην εφαρμογή προστέθηκε η δυνατότητα Google Sign In για είσοδο με τα credential που ο χρήστης έχει δώσει στη google για είσοδο στις υπηρεσίες της. Αυτό πραγματοποιείται με τον κώδικα που βρίσκεται στις κλάσεις SignInActivity.java, GoogleAuth.java και LoginWithGoogleSignIn.java . Τα διαδοχικά βήματα που ακολουθούνται φαίνονται στις παρακάτω εικόνες:

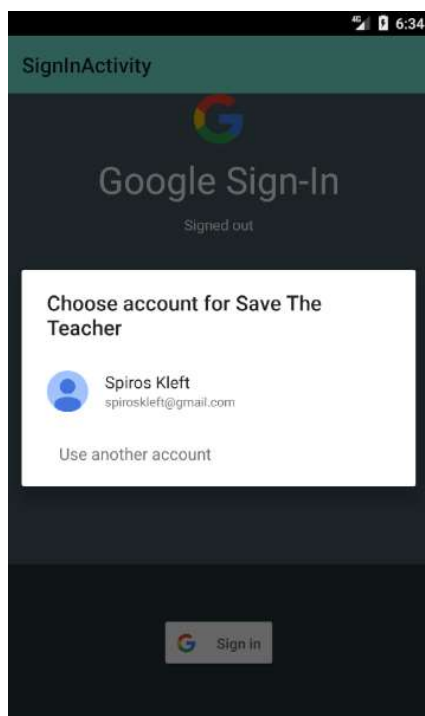
Αρχικά επιλέγουμε το “Google Sign In “ κουμπί,



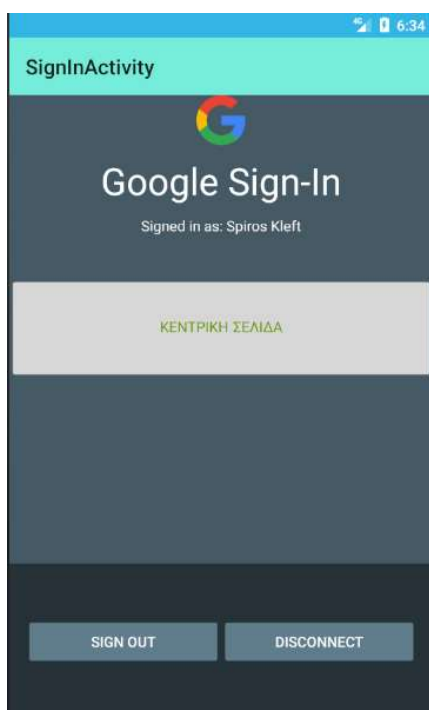
Επιλέγουμε αν είμαστε “signed out” το κουμπί “Sign in”



Επιλέγουμε τον λογαριασμό που επιθυμούμε να συνδέσουμε στην εφαρμογή,

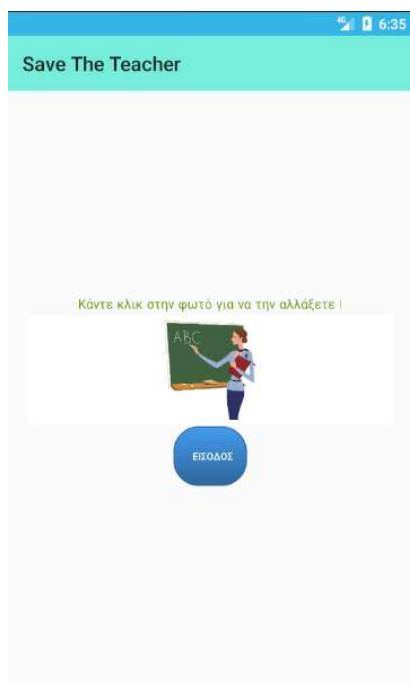


Επιλέγουμε να συνεχίσουμε ή να αποσυνδεθούμε



Επιλέγουμε αν το επιθυμούμε να προσθέσουμε μία εναλλακτική φωτογραφία στο προφίλ μας ή να κρατήσουμε την default.





### 3.6. Firebase Notifications

Χρησιμοποιήθηκαν τα Firebase notifications και συγκεκριμένα το cloud messaging προκειμένου να στέλνονται διάφορες ειδοποιήσεις στους χρήστες της εφαρμογής όταν αυτή είναι στο παρασκήνιο. Τούτο επιτεύχθηκε με τον κώδικα που βρίσκεται στις κλάσεις `MyFirebaseMessagingService.java` και `MyFirebaseInstanceIdService.java`. Επίσης στέλνονται οι ειδοποιήσεις που επιθυμούμε

Message	Status	Delivery date	Platform	Target estimate	Open rate
Koko brodo users tou Save the tea...	✓ Completed	Jun 22, 2017 11:32 PM	Android	<1000	33%
Panella is superman	✓ Completed	Jun 20, 2017 8:15 PM	Android	<1000	25%
gagag GERGGERRRRRRR	✓ Completed	Jun 20, 2017 12:50 PM	Android	<1000	17%
Τσουτσουτάκι !!! Βγήκαν οι Ανοηληστές !!!	✓ Completed	Jun 19, 2017 9:49 PM	Android	<1000	14%
Superman	✓ Completed	Jun 19, 2017 9:48 PM	iOS	<1000	100%
fafafa fafafafafa	✓ Completed	Jun 19, 2017 9:46 PM	iOS	<1000	—
wake uprrrrr	✓ Completed	Jun 19, 2017 9:46 PM	iOS	<1000	—



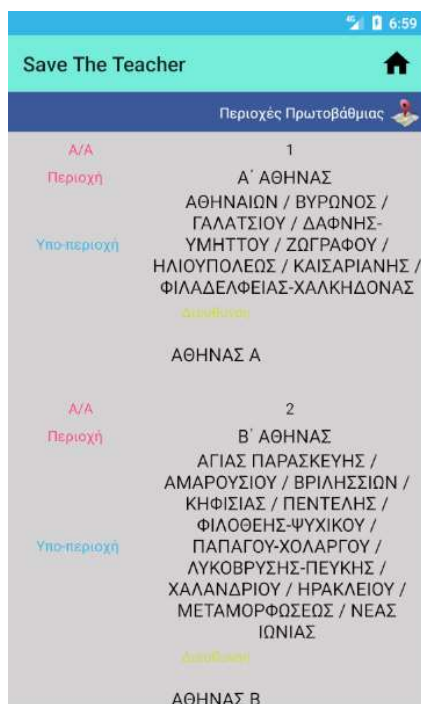
### 3.7. Realm

Στην παρούσα εφαρμογή χρησιμοποιήθηκε το Realm προκειμένου να πάρουμε ένα json αρχείο και να το μετατρέψουμε σε Realm βάση δεδομένων. Τούτο επιτυγχάνεται με τις κλάσεις RegionsRealm.java, Region.java, RegionsListActivity.java

Έτσι έχουμε το αρχείο json με όλες τις περιοχές των αναπληρωτών πρωτοβάθμιας εκπαίδευσης στην Ελλάδα όπως φαίνεται παρακάτω:

```
{
  {
    "Id": 1,
    "Name": "Α' ΑΘΗΝΑΣ",
    "subName": "ΑΘΗΝΑΙΩΝ / ΒΥΡΩΝΟΣ / ΓΑΛΑΤΣΙΟΥ / ΔΑΦΝΗΣ-ΥΜΗΤΤΟΥ / ΕΣΤΡΑΦΟΥ / ΗΛΙΟΥΠΟΛΕΩΣ / ΚΑΙΣΑΡΙΑΝΗΣ / ΦΙΛΑΔΕΛΦΕΙΑΣ-ΧΑΛΚΗΔΟΝΑΣ",
    "Office": "ΑΘΗΝΑΣ Α"
  },
  {
    "Id": 2,
    "Name": "Β' ΑΘΗΝΑΣ",
    "subName": "ΑΓΙΑΣ ΠΑΡΑΣΚΕΥΗΣ / ΑΜΑΡΟΥΣΙΟΥ / ΒΡΙΛΗΣΙΩΝ / ΚΗΦΙΣΙΑΣ / ΠΕΝΤΕΛΗΣ / ΦΙΛΟΘΕΗΣ-ΨΥΧΙΚΟΥ / ΠΑΠΑΓΟΥ-ΧΟΛΑΡΟΥ / ΛΥΚΟΒΥΣΣΗΣ-ΠΕΥΚΗΣ / ΧΑΛΑΝΔΡΙΟΥ / ΗΡΑΚΛΕΙΟΥ / ΜΕΤΑΜΟΡΦΩΣΕΩΣ /",
    "Office": "ΑΘΗΝΑΣ Β"
  },
  {
    "Id": 3,
    "Name": "Γ' ΑΘΗΝΑΣ",
    "subName": "ΑΙΓΑΛΕΩ / ΑΓΙΑΣ ΒΑΡΒΑΡΑΣ / ΧΑΪΔΑΡΙΟΥ / ΠΕΡΙΣΤΕΡΙΟΥ / ΑΓΙΩΝ ΑΝΑΡΤΥΩΝ-ΚΑΜΑΤΕΡΟΥ / ΙΛΙΟΥ / ΠΕΤΡΟΥΠΟΛΕΩΣ",
    "Office": "ΑΘΗΝΑΣ Γ"
  },
  {
    "Id": 4,
    "Name": "Δ' ΑΘΗΝΑΣ",
    "subName": "ΑΓΙΟΥ ΔΗΜΗΤΡΙΟΥ / ΑΛΙΜΟΥ / ΕΛΛΗΝΙΚΟΥ-ΑΡΤΥΡΟΠΟΛΗΣ / ΤΑΥΦΑΔΑΣ / ΚΑΛΛΙΘΕΑΣ / ΠΑΛΑΙΟΥ ΦΑΛΗΡΟΥ / ΝΕΑΣ ΣΜΥΡΝΗΣ / ΜΟΣΧΑΤΟΥ-ΤΑΥΡΟΥ",
    "Office": "ΑΘΗΝΑΣ Δ"
  },
  {
    "Id": 5,
    "Name": "ΑΙΤΩΛΟΚΑΡΦΑΝΙΑΣ",
    "subName": "ΝΟΜΟΣ ΑΙΤΩΛ/ΝΙΑΣ",
    "Office": "ΑΙΤΩΛΟΚΑΡΦΑΝΙΑΣ"
  },
  {
    "Id": 6,
    "Name": "Α' ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ",
    "subName": "ΚΡΕΜΙΔΑΣ / ΜΑΡΚΟΠΟΥΛΟΥ ΜΕΣΟΓΑΙΑΣ / ΠΑΙΔΙΑΣ / ΛΑΥΡΕΩΤΙΚΗΣ / ΣΠΑΤΩΝ-ΑΡΤΕΜΙΔΟΣ / ΣΑΡΩΝΙΚΟΥ / ΠΑΛΛΗΝΗΣ / ΡΑΦΗΝΑΣ-ΠΙΚΕΡΜΙΟΥ / ΜΑΡΑΘΩΝΟΣ / ΒΑΦΗΣ-ΒΟΥΛΑΣ-ΒΟΥΛΙΑΓΜΕΝΗΣ",
    "Office": "ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ"
  },
  {
    "Id": 7,
    "Name": "Β' ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ",
    "subName": "ΟΡΓΟΠΟΥ / ΔΙΟΝΥΣΟΥ / ΑΧΑΡΝΩΝ",
    "Office": "ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ"
  },
}
```

Στη συνέχεια πατώντας στο μενού τις εφαρμογής το κουμπί «Περιοχές» αυτομάτως με τις κατάλληλες μεθόδους φτιάχνουμε μία βάση τοπικά σε Realm την οποία στην συνέχεια καλούμε για να απεικονίσουμε τις περιοχές.

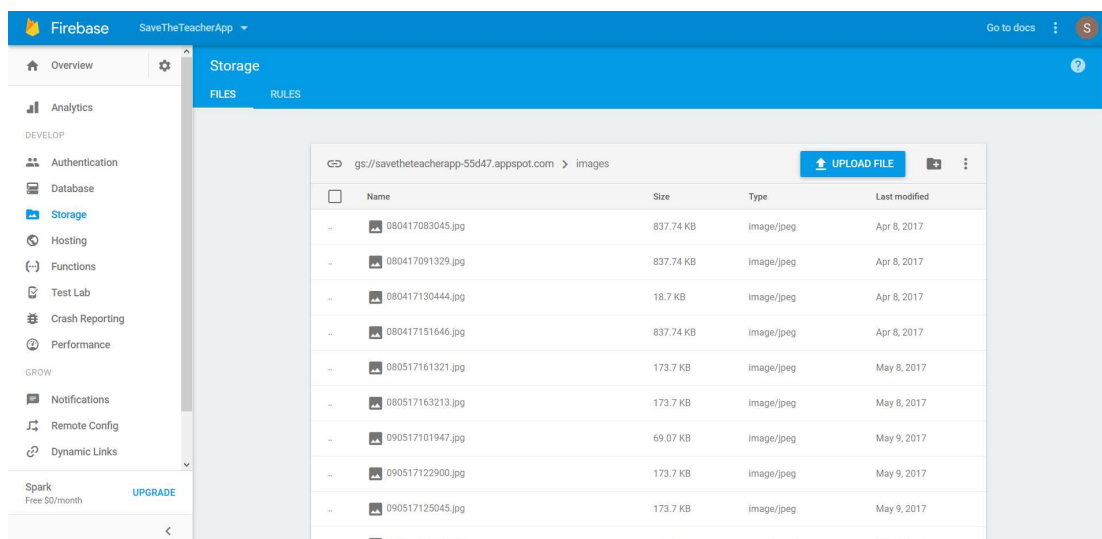


### 3.8. Picasso

Χρησιμοποιούμε το Picasso (<http://square.github.io/picasso/>) απλά για το handle των φωτογραφιών σε κάθε δημοσίευση «σκέψης»

### 3.9. Firebase Storage

Χρησιμοποιούμε στην εφαρμογή την δυνατότητα που μας δίνει το Firebase για αποθήκευση αρχείων και συγκεκριμένα στην εφαρμογή μας για αποθήκευση των φωτογραφιών κάθε χρήστη, είτε τις φωτογραφίες προφίλ είτε τις φωτογραφίες των «σκέψεων» του.



Συγκεκριμένα όπως προαναφέραμε όλες οι φωτογραφίες αποθηκεύονται εδώ στο φάκελο /images/ και στη βάση sql αποθηκεύουμε μόνο το link των φωτογραφιών στο firebase storage. Τα παραπάνω υλοποιούνται στην MainActivity.java και συγκεκριμένα στη μέθοδο uploadimage() ως εξής:

```
FirebaseStorage storage=FirebaseStorage.getInstance();  
// Create a storage reference from our app  
StorageReference storageRef =  
storage.getReferenceFromUrl("gs://savetheteacherapp-  
55d47.appspot.com");  
DateFormat df = new SimpleDateFormat("ddMMyyHHmmss");  
Date dateobj = new Date();  
  
String mydownloadUrl=SaveSettings.UserID+ "_" + df.format(dateobj)  
+".jpg";  
StorageReference mountainsRef = storageRef.child("images/"+  
mydownloadUrl);
```

Αναλυτικότερα ανατρέξτε στον κώδικα της εφαρμογής:  
<https://github.com/Spiroskleft/SaveTheTeacher/blob/master/app/src/main/java/com/example/spiros/savetheteacher/MainActivity.java>

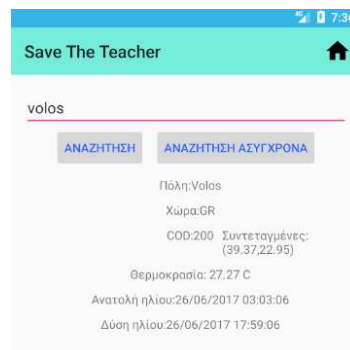
### 3.10. Retrofit API

Χρησιμοποιήθηκε το Retrofit στην εφαρμογής μας προκειμένου να λαμβάνεται από το openweather η πρόγνωση του καιρού τοπικά (αναλόγως με την τοποθεσία) για επτά ημέρες καθώς και για οποιαδήποτε περιοχή στον κόσμο αυτή τη στιγμή με στοιχεία συντεταγμένων θερμοκρασίας και ανατολής, δύσης ηλίου. Συγκεκριμένα χρησιμοποιήθηκε το <http://api.openweathermap.org> καθώς και:



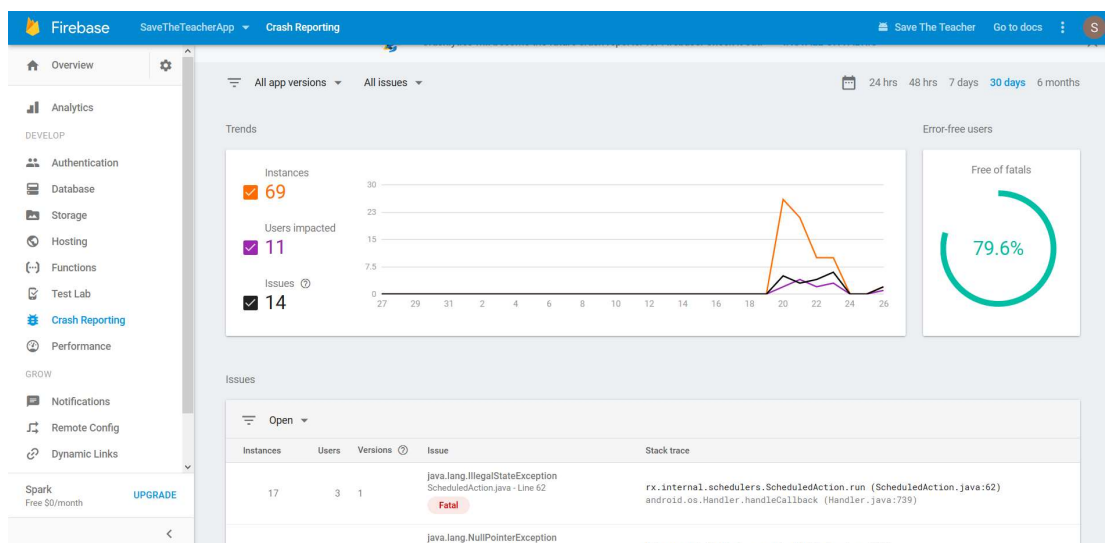
```
public interface WeatherApi {  
    @GET("/data/2.5/weather")  
    void getWeatherFromApi (  
        @Query("q") String cityName,  
        @Query("APPID") String appId,  
        Callback<WeatherData> callback);  
    @GET("/data/2.5/weather")  
    WeatherData getWeatherFromApiSync (  
        @Query("q") String cityName,  
        @Query("APPID") String appId);  
}
```

Οι κλάσεις που υλοποιούν τα παραπάνω βρίσκονται εντός του φακέλου Weather στον κώδικά και συγκεκριμένα:  
<https://github.com/Spiroskleft/SaveTheTeacher/tree/master/app/src/main/java/com/example/spiros/savetheteacher/Weather>



### 3.11. Firebase Crash Reporting

Η εφαρμογή στάλθηκε ως .apk σε αριθμό φίλων-users προκειμένου να «τρέξει» σε όσο το δυνατό μεγαλύτερο αριθμό διαφορετικών συσκευών. Με τη χρήση του Firebase Crash Reporting αυτές συγκεντρώνονται και οπτικοποιούνται προκειμένου να έχουμε μία πολύ καλή εικόνα των σφαλμάτων που εμπεριέχει ο κώδικας της εφαρμογής.



Παραπάνω αναφέρθηκαν οι κυριότερες τεχνολογίες που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής. Λεπτομερώς ο κώδικας μπορεί να βρεθεί στο github: <https://github.com/Spiroskleft/SaveTheTeacher> .

## 4. Σύντομος Οδηγός Χρήστη

Παρακάτω θα παρουσιάσουμε τις βασικές λειτουργίες και τον τρόπο χρήσης της εφαρμογής εν συντομία.

ΠΡΟΣΟΧΗ: ΘΑ ΠΡΕΠΕΙ ΝΑ ΜΠΟΡΕΙΤΕ ΝΑ ΧΤΥΠΗΣΕΤΕ ΤΟ:  
<http://83.212.99.161:8083/twitterserver/userfollowing.php>  
ΓΙΑ ΝΑ ΤΡΕΞΕΙ ΣΩΣΤΑ Η ΕΦΑΡΜΟΓΗ. (για κάποιο λόγο από το πανεπιστήμιο δεν επιτρέπεται να χτυπήσουμε τον ~okeanos)

### 4.1. Είσοδος στην εφαρμογή

Όπως αναφέρθηκε και παραπάνω υπάρχουν δύο εναλλακτικά Login/Register στην εφαρμογή μας. Ο πρώτος με δικά μας credentials και ο δεύτερος με Google Sign In όπως φαίνεται και στην εικόνα:



Save The Teacher

Επιλέξτε τον τρόπο σύνδεσης στην εφαρμογή

επίλεξε username

επίλεξε email

επίλεξε password

LOGIN

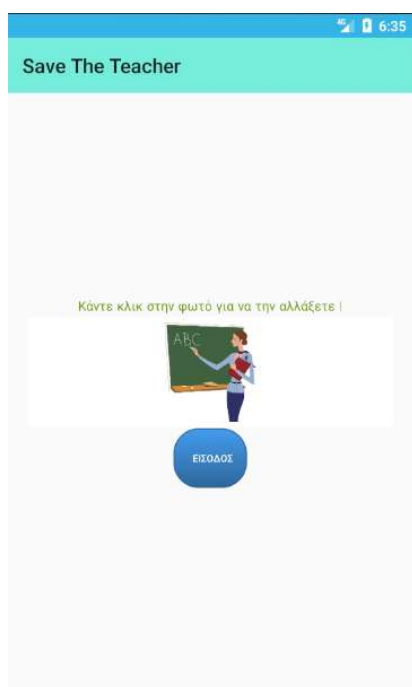
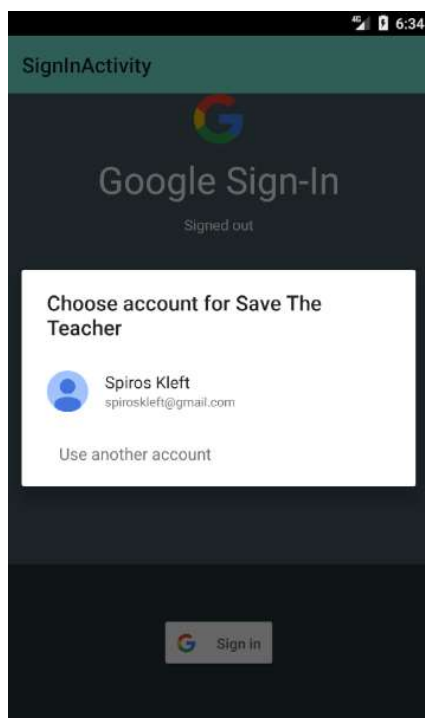
OR

GOOGLE SIGN IN

Φυσικά μπορούμε να αλλάξουμε και την εικόνα του προφίλ μας απλά πατώντας επάνω στην φωτογραφία. Εάν επιλέξουμε Google Sign In την φωτογραφία την αλλάζουμε σε επόμενη σελίδα μετά το signed in.







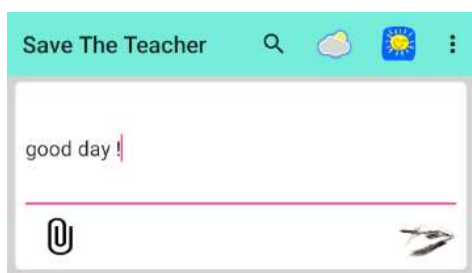
#### 4.2. Κύρια Οθόνη Εφαρμογής

Με την επιτυχή σύνδεσή μας εμφανίζεται το μήνυμα “user is added” και ο χρήστης πλέον έχει καταχωρηθεί στη ΒΔ. Σε κάθε νέο άνοιγμα της εφαρμογής δεν χρειάζεται εκ νέου login φυσικά και ο χρήστης μεταφέρεται κατευθείαν στην αρχική σελίδα της εφαρμογής



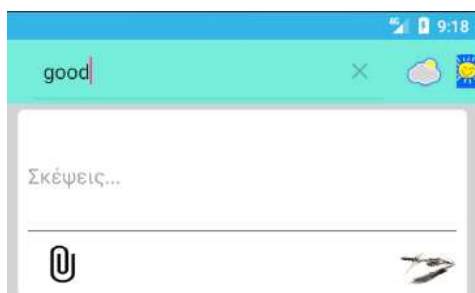
Στην παραπάνω εικόνα φαίνεται η αρχική σελίδα ενός χρήστη που μόλις μπήκε για πρώτη φορά στην εφαρμογή και δεν έχει κάνει post καμία του «σκέψη» αλλά και ούτε ακολουθεί κάποιον άλλο χρήστη της εφαρμογής.

Εδώ, λοιπόν, ο χρήστης μπορεί είτε να ποστάρει τη δική του «σκέψη»



πληκτρολογώντας την και προσθέτοντας αν επιθυμεί φωτογραφία πατώντας τον συνδετήρα και στη συνέχεια το χέρι με τη πένα

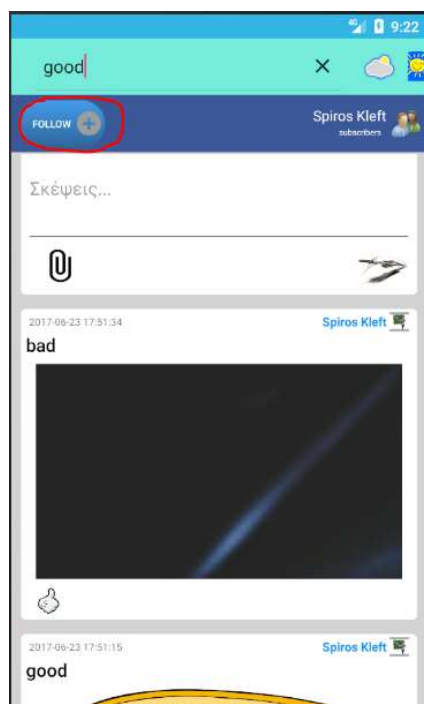
είτε να αναζητήσει «σκέψεις» άλλων χρηστών πατώντας τον μεγεθυντικό φακό στο μενού και πληκτρολογώντας λέξεις που μπορεί να τον ενδιαφέρουν.



Αν βρεθεί «σκέψη» με αυτή τη λέξη κλειδί θα εμφανιστεί στη κύρια οθόνη. Αν ο χρήστης πατήσει επάνω στον χρήστη που έχει κάνει αυτή την «σκέψη»



θα μεταφερθεί στη σελίδα του προφίλ του χρήστη αυτού και θα έχει τη δυνατότητα αν επιθυμεί να ακολουθεί τον χρήστη αυτών και να βλέπει κάθε «σκέψη» του όταν ανοίγει την εφαρμογή.




Φυσικά αν στο μέλλον δεν επιθυμεί να ακολουθεί αυτόν τον χρήστη μπορεί απλά από την ίδια σελίδα να τον κάνει unfollow !



Στη συνέχεια από το μενού επιλέγοντας την αρχική σελίδα θα μεταφερθεί ο χρήστης στη κύρια σελίδα όπου πλέον θα μπορεί να βλέπει όλες τις «σκέψεις» όλων όσων έχει επιλέξει να ακολουθεί.




#### 4.3. Καιρός στον Κόσμο

Από το μενού επιλογών της εφαρμογής πατώντας επάνω στο εικονίδιο  εμφανίζεται η σελίδα του καιρού ανά τον κόσμο αναλόγως της πόλης που εμείς αναζητούμε.



Σε αυτή τη σελίδα μπορούμε να δούμε πληροφορίες που σχετίζονται με τις συντεταγμένες, τη θερμοκρασία και την ανατολή και δύση του ηλίου στη πόλη προς αναζήτηση που έχουμε επιλέξει.

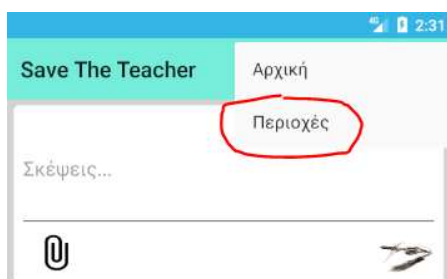
#### 4.4. Πρόγνωση Καιρού

Πατώντας στο εικονίδιο του μενού της εφαρμογής  εμφανίζεται η σελίδα της πρόγνωσης εφτά ημερών της τοποθεσίας του χρήστη.

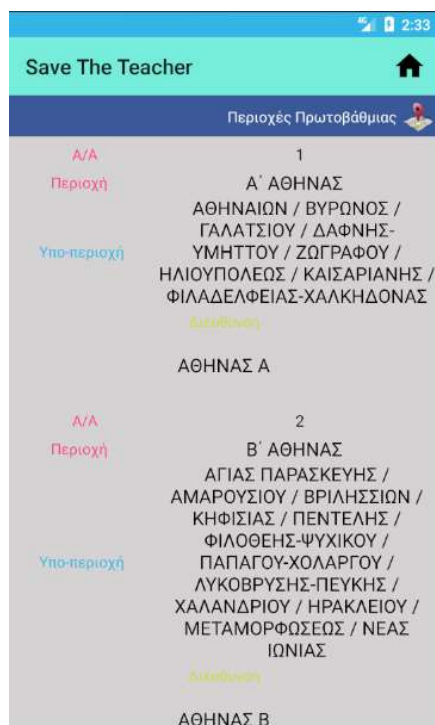


#### 4.5. Περιοχές Διευθύνσεων Πρωτοβάθμιας Εκπαίδευσης

Πατώντας στο μενού επιλογών της εφαρμογής στην επιλογή «Περιοχές»



θα εμφανιστεί η σελίδα με όλες τις δυνατές περιοχές που κάποιος αναπληρωτής δάσκαλος μπορεί να πάει ως τοποθέτηση ανά την Ελλάδα όπως φαίνεται στην παρακάτω εικόνα.



Εδώ μπορεί ο χρήστης να δει την περιοχή, τις υπο-περιοχές διορισμού καθώς και τη Διεύθυνση Πρωτοβάθμιας εκπαίδευσης που ανήκει.

#### 4.6. Λοιπά Χαρακτηριστικά

Τέλος υπάρχει μία σειρά από διασκεδαστικά χαρακτηριστικά που προστέθηκαν στην εφαρμογή προκειμένου να γίνει πιο ευχάριστη. Για παράδειγμα ένα απλό μήνυμα αν κάποιος πατήσει να δημοσιεύσει μία σκέψη χωρίς να βάλει κάποιο κείμενο (Δοκιμάστε το!). Ένα άλλο χαρακτηριστικό είναι αναλόγως της ώρας της ημέρας που ο χρήστης χρησιμοποιεί την εφαρμογή λαμβάνει μηνύματα χαιρετισμού ! (Η παραπάνω μέθοδος ονομάζεται *kalimeres()* στο κώδικα στη *MainActivity.java* κλάση).

### 5. Συμπεράσματα – Προβλήματα – Μελλοντική Εργασία

Κατά την διάρκεια της ανάπτυξης της εφαρμογής επιλύθηκαν διάφορα θέματα που συναντήθηκαν επιτυχώς. Συναρτήσε του χρόνου που δόθηκε για την ανάπτυξη της εφαρμογής καθώς και του γεγονότος ότι για τον γράφοντα ήταν η πρώτη εφαρμογή σε Android πιθανώς η εφαρμογή δεν είναι απολύτως ολοκληρωμένη. Για παράδειγμα σε μελλοντικό χρόνο ο χρήστης θα πρέπει να επιλέγει την περιοχή που διορίστηκε ως αναπληρωτής και να μπορεί κάνοντας *search* να βρίσκει όλους όσους έχουν διοριστεί εκεί. Επίσης στο *search* πρέπει





να προστεθεί η λειτουργία έρευνας και με βάσει το username και όχι μόνο των «σκέψεων». Επιπρόσθετα μπορούν να προστεθούν λειτουργίες chat μεταξύ των χρηστών. Τέλος η πλήρης ασφάλεια της εφαρμογής καθώς και η πλήρης αποσφαλματωσή της αποτελούν μελλοντικές εργασίες που στα πλαίσια του χρόνου που υπήρχε στη διάθεση μας δεν κατέστησαν δυνατά.