



## Εφαρμογή για Android “RememberToDo”

## Android application “RememberToDo”

Διπλωματική Εργασία

ΤΟΥ

**Καραναστάση Δημήτρη**

**Υπεύθυνος Καθηγητής:** Αλκιβιάδης Γ. Ακρίτας

**Καθηγητής Π.Θ.**

**Επιβλέπων Καθηγητής:** Γεώργιος Σταμούλης

**Καθηγητής Π.Θ.**



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

## **Εφαρμογή για Android “RememberToDo”**

### **Android application “RememberToDo”**

Διπλωματική Εργασία

ΤΟΥ

**Καραναστάση Δημήτρη**

Εγκρίθηκε από την διμελή επιτροπή την ημερομηνία εξέτασης

*(Υπογραφή)*

*(Υπογραφή)*

.....

.....

ΚΥΡΙΟΣ ΕΠΙΒΛΕΠΩΝ

ΔΕΥΤΕΡΕΥΩΝ ΕΠΙΒΛΕΠΩΝ

Καθηγητής Π.Θ.

Καθηγητής Π.Θ.

(Υπογραφή)

.....

**Καραναστάσης Δημήτριος**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών,  
Πανεπιστημίου Θεσσαλίας

© 2015 – All rights reserved

Βόλος, Μάρτιος 2015

## Στην οικογένεια μου και στους φίλους μου...

### Ευχαριστίες

Με την περάτωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες της Διπλωματικής εργασίας κ. Αλκιβιάδη Ακρίτα και κ. Γεώργιο Σταμούλη για την εμπιστοσύνη που επέδειξαν στο πρόσωπό μου, για την άριστη συνεργασία, τις ουσιώδεις υποδείξεις και παρεμβάσεις, που διευκόλυναν την εκπόνηση της Διπλωματικής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση και υποστήριξη της καθ' όλη την διάρκεια της ζωής μου καθώς και τους φίλους μου που στάθηκαν δίπλα μου σε όλα τα φοιτητικά χρόνια.

**Καραναστάσης Δημήτριος**

**Βόλος, 2015**

## Περίληψη

Σκοπός της παρούσας Διπλωματικής εργασίας ήταν η σχεδίαση και υλοποίηση μιας εφαρμογής στο Android λειτουργικό η οποία στόχο έχει να βοηθήσει τον χρήστη στην καθημερινότητά του καθώς θα του υπενθυμίζει οτιδήποτε έχει γράψει την ημερομηνία και ώρα ή στην θέση στον χάρτη που έχει επιλέξει.

Με τους ρυθμούς που κινείται σήμερα η ζωή μας είναι λογικό να ξεχνάμε διάφορα πράγματα που θέλουμε να κάνουμε. Η λύση πριν λίγα χρόνια για αυτό ήταν να τα σημειώνουμε κάπου. Με την εξέλιξη όμως της τεχνολογίας και την εμφάνιση και εξέλιξη των smartphones τα πράγματα γίνονται ευκολότερα. Στην παρούσα εφαρμογή ο χρήστης μπορεί να σημειώσει ότι θέλει να του υπενθυμίσει το smartphone του και να επιλέξει την ημερομηνία ή την τοποθεσία στον παγκόσμιο χάρτη που θέλει να ειδοποιηθεί. Μόλις λοιπόν φτάσει η ώρα ή το μέρος που έχει επιλέξει τότε εμφανίζεται μια ειδοποίηση με τον τίτλο και την περιγραφή της υπενθύμισης που είχε δημιουργήσει. Πατώντας την ο χρήστης βλέπει αναλυτικά την υπενθύμιση που είχε φτιάξει και μπορεί να την διαγράψει ή να την επαναλάβει.

Με αυτό λοιπόν τον τρόπο η εφαρμογή αυτή είναι χρήσιμη για άτομα που συνεχώς ξεχνάνε γεγονότα και θέλουν να τα θυμηθούν συγκεκριμένη ώρα ή σε συγκεκριμένο μέρος βοηθώντας τους στην ομαλή πορεία της καθημερινής ρουτίνας.

## Abstract

The purpose of this thesis was to design and implement an application on Android operating which aims to assist the user in everyday life as it reminds him of everything he wants by date and time or position on the map where is selected.

With the rate life moves, today is reasonable to forget things we want to do. The solution a few years ago was to note them somewhere. But with the evolution of technology and the evolution of smartphones things are easier. In this application the user can note what he wants the smartphone to remind him and select the date or the location on the world map he wants to be notified. So when the time arrives or at the place he has chosen then a notification is displayed with the title and the

description of the reminder that he had created. By clicking on it the user sees the reminder that she had created, and may delete or re-set it.

So this way the application is useful for people who constantly forget events and want to remember them on a specific time or a specific place and makes the course of daily life a lot smoother.

## Περιεχόμενα

<b>1. Εισαγωγή.....</b>	<b>10</b>
1.1 Αρχική ιδέα.....	10
1.2 Γενικά για το Android.....	11
1.3 Ιστορική Αναδρομή.....	12
1.4 Εκδόσεις Android.....	14
1.5 Αρχιτεκτονική Android.....	16
<b>2. Προγραμματισμός στο Android .....</b>	<b>18</b>
2.1 Android SDK.....	18
2.2 Google Play Services.....	19
2.2.1 Google Maps.....	19
2.2.2 Google Places.....	22
2.3 Android Emulator.....	24
2.3.1 Gevymotion Emulator .....	26
<b>3. Περιεχόμενα Εφαρμογής Android.....</b>	<b>29</b>
3.1 Βασικές Πληροφορίες.....	29
<b>4. Επίδειξη Εφαρμογής RememberToDo.....</b>	<b>47</b>
4.1 Βασικές λειτουργίες της εφαρμογής.....	47
4.1.1 Ειδοποίηση Ημερομηνίας (Date & Time Notification).....	49
4.1.2 Ειδοποίηση Τοποθεσίας (GPS Notification).....	50
4.1.3 Συγχρονισμός ειδοποιήσεων με Ημερολόγιο (Calendar Sync) .....	53
4.2 Ειδοποίηση χρήστη από την εφαρμογή.....	54
4.3 Ρυθμίσεις εφαρμογής.....	55
<b>5. Τελικές Σκέψεις - Επίλογος.....</b>	<b>56</b>
<b>6. Βιβλιογραφία .....</b>	<b>57</b>

## Ευρετήριο Εικόνων

Εικόνα 1. Το λογότυπο του Android .....	11
Εικόνα 2. Το πρώτο Android smartphone T-Mobile G1 .....	12
Εικόνα 3. Motorola Droid .....	13
Εικόνα 4. Android Market .....	13
Εικόνα 5. Android 1.5 Cupcake .....	14
Εικόνα 6. Widgets στην homescreen .....	14
Εικόνα 7. Android 1.6 Donat .....	14
Εικόνα 8. Android 2.2 Froyo .....	14
Εικόνα 9. Google Nexus One .....	14
Εικόνα 10. Android 2.3 Gingerbread .....	14
Εικόνα 11. Διαμάχη μεταξύ Android και iOS .....	15
Εικόνα 12. Android 3.0 HoneyComb .....	15
Εικόνα 13. Android 4.0 Ice Cream Sandwich .....	15
Εικόνα 14. Android 4.4 Kit Kat .....	15
Εικόνα 15. Android 4.2 Jelly bean .....	15
Εικόνα 16. Διάγραμμα αρχιτεκτονικής του Android .....	16
Εικόνα 17. Λειτουργία του Android .....	17
Εικόνα 18. Eclipse με εγκατεστημένο το Android SDK .....	18
Εικόνα 19. Google Play Services .....	19
Εικόνα 21. Κανονική, Δορυφορική και Υβριδική (Terrain) εμφάνιση Google Maps .....	20
Εικόνα 20. Google Maps .....	20
Εικόνα 22. Zoom Controls, Compass και My Location Button .....	21
Εικόνα 23. Πατώντας το My Location Button εμφάνιση Marker και στο InfoWindow η πόλη .....	22
Εικόνα 24. Marker και InfoWindow για μια περιοχή ειδοποίησης στον Βόλο .....	22
Εικόνα 25. Υπηρεσία Google Places .....	23
Εικόνα 26. Autocomplete τοποθεσίας στον χάρτη .....	23
Εικόνα 27. Αναζήτηση της τοποθεσίας στο autocomplete με Google Places .....	23
Εικόνα 28. Τοποθέτηση πινέζας στην επιλεγμένη τοποθεσία .....	23
Εικόνα 29. AVD Manager στο Eclipse .....	24
Εικόνα 31. Android Emulator .....	25
Εικόνα 30. Προσθήκη νέας συσκευής (Emulator) στο Eclipse .....	25
Εικόνα 32. Ιστοσελίδα της Gevymotion .....	26
Εικόνα 33. Επιλογή συσκευής και έκδοσης Gevymotion Emulator .....	27
Εικόνα 34. Nexus 7 Emulator .....	27
Εικόνα 35. Εικονική επεξεργασία της μπαταρίας, της θέσης στο GPS και της κάμερας από το Gevymotion Emulator .....	28
Εικόνα 36. Οι φάκελοι και υποφάκελοι της εφαρμογής που περιέχουν τα xml αρχεία και τα Activities όπως φαίνονται στο project manager του Eclipse .....	29
Εικόνα 37. Δημιουργία xml αρχείου στο Eclipse και τα διάφορα στοιχεία που μπορεί να χρησιμοποιήσει ο developer .....	30
Εικόνα 38. Κώδικας xml που ορίζει τα χαρακτηριστικά κάθε στοιχείου που υπάρχει μέσα στο datenotification xml .....	31
Εικόνα 39. Διάγραμμα ζωής Activity .....	33
Εικόνα 40. Επαναφορά – Αποθήκευση προηγούμενης κατάστασης Activity .....	34
Εικόνα 41. Στοίβα των Activities και navigation με το Back Button .....	35
Εικόνα 42. Παράδειγμα χρήσης Service από την εφαρμογή για την ειδοποίηση Date and Time .....	36



Εικόνα 43. Ειδοποιήσεις στο notification drawer.....	37
Εικόνα 44. Εμφάνιση ειδοποίησης στη notification area .....	37
Εικόνα 45. Control Widget .....	42
Εικόνα 46. Collection Widget.....	42
Εικόνα 47. Information Widget .....	42
Εικόνα 48. Widget με ειδοποίηση ημερομηνίας.....	47
Εικόνα 49. Widget με ειδοποίηση τοποθεσίας .....	47
Εικόνα 50. Αρχική οθόνη RememberToDo.....	47
Εικόνα 51. Δημιουργία νέας ειδοποίησης .....	48
Εικόνα 52. Επιλογή τύπου ειδοποίησης.....	49
Εικόνα 53. Δημιουργία νέας ειδοποίησης Ημερομηνίας κ' Ώρας.....	50
Εικόνα 54. Περιγραφή – Επεξεργασία – Διαγραφή ειδοποίησης Ημερομηνίας κ' Ώρας .....	50
Εικόνα 55. Ειδοποίηση τοποθεσίας με αναζήτηση.....	51
Εικόνα 56. Χάρτες Google για ειδοποίηση τοποθεσίας .....	51
Εικόνα 57. Δημιουργία ειδοποίησης τοποθεσίας.....	52
Εικόνα 58. Προειδοποιητικό μήνυμα διακοπής του GPS όταν υπάρχει ειδοποίηση τοποθεσίας .....	52
Εικόνα 59. Προειδοποιητικό μήνυμα ότι το GPS λειτουργεί ξανά .....	53
Εικόνα 60. Συγχρονισμένη ειδοποίηση με Google Calendar .....	53
Εικόνα 61. Συγχρονισμένη ειδοποίηση με το Ημερολόγιο .....	53
Εικόνα 62. Λίστα συγχρονισμού ειδοποιήσεων ημερομηνίας.....	53
Εικόνα 63. Ειδοποίηση από την εφαρμογή στην μπάρα της συσκευής.....	54
Εικόνα 64. Ειδοποίηση από Google Calendar και την εφαρμογή Ημερολόγιο.....	54
Εικόνα 65. Ειδοποίηση ημερομηνίας.....	54
Εικόνα 66. Ειδοποίηση τοποθεσίας από την εφαρμογή στην μπάρα της συσκευής ...	55
Εικόνα 67. Ειδοποίηση τοποθεσίας .....	55

# 1. Εισαγωγή

## 1.1 Αρχική ιδέα

Το εαρινό εξάμηνο του 2012-2013 παρακολούθησα το μάθημα “Ανάπτυξη εφαρμογών σε iOS περιβάλλον” του κ. Αλκιβιάδη Ακρίτα σαν ειδικό θέμα. Ενθουσιάστηκα πάρα πολύ με το αντικείμενο του προγραμματισμού εφαρμογών σε iOS τόσο που ήθελα να δοκιμάσω και κάτι καινούριο. Για τον λόγο αυτό αποφάσισα να φτιάξω μια εφαρμογή σε Android πλατφόρμα για να ειδικευτώ και σε αυτό τον τομέα.

Μετά ακολούθησε ένα ταξίδι μέσα από το οποίο απέκτησα αρκετές προγραμματιστικές γνώσεις πάνω στο αντικείμενο και εξοικειώθηκα αρκετά με το Eclipse και τους simulators που χρησιμοποιούνται για το Android Development. Φυσικά υπήρξαν και πολλά εμπόδια όμως ψάχνοντας στο ιντερνέτ και επικοινωνώντας με άτομα που είχαν το ίδιο πάθος για development όλα λύθηκαν και τελικά κατάφερα να ολοκληρώσω επιτυχώς την εφαρμογή.

Σκοπός μου ήταν να βοηθήσω τους ανθρώπους που δυσκολεύονται να θυμούνται διάφορα πράγματα και γεγονότα ,δεν θα άφηνα τον εαυτό μου εκτός φυσικά, δίνοντάς τους την δυνατότητα να τα σημειώνουν στην εφαρμογή και αυτή να τους τα θυμίζει όποτε αυτοί το έχουν ορίσει. Η όλη ομορφιά της βρίσκεται στο γεγονός ότι κάνει κάποια πράγματα πέρα από τις άλλες κοινές εφαρμογές υπενθύμισης – ατζέντα που ήδη υπάρχουν σε όλα τα smartphones, χρησιμοποιεί και το GPS του κινητού για να μπορεί ο χρήστης να ορίσει και το μέρος που θα ήθελε να ειδοποιηθεί. Έτσι δεν χρειάζεται να σκέφτεται την ώρα που θέλει να ειδοποιηθεί αν δεν τον ενδιαφέρει, αλλά το μέρος ώστε να ειδοποιείται έγκυρα και όχι σκόπιμα. Τέλος υπάρχει και η δυνατότητα του συγχρονισμού με το ημερολόγιο – ατζέντα του smartphone ακόμα και με το Google Calendar, αν αυτό είναι διαθέσιμο, κάνοντας την εφαρμογή ιδιαίτερα αξιόπιστη.

Στις παρακάτω σελίδες παρουσιάζεται η πλατφόρμα Android και τα διαφορετικά στάδια που ακολούθησα για την ανάπτυξή της εφαρμογής.

## 1.2 Γενικά για το Android

Το Android είναι ένα λειτουργικό σύστημα που ενσωματώνεται σε συσκευές κινητής τηλεφωνίας, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα kernel του λειτουργικού Linux και ακόμη, επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού της Google. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.



Εικόνα 1. Το λογότυπο του Android

Συσκευές με Android υπάρχουν πλέον πάρα πολλές, η καθεμία με διαφορετικά χαρακτηριστικά και από διάφορες κατασκευάστριες εταιρίες όπως LG, Samsung, HTC, Sony, Motorola, είναι μερικές από τις εταιρίες που χρησιμοποιούν το λειτουργικό Android για τα smartphones τους. Το πολύ θετικό με τις συσκευές Android είναι ότι είναι αφενός multimedia συσκευές, δηλαδή σας δίνουν τη δυνατότητα να αναπαράγετε πολλαπλά μέσα και multitasking, δηλαδή σας δίνουν τη δυνατότητα εκτέλεσης πολλών εφαρμογών ταυτόχρονα, π.χ. ακούτε τραγούδια ενώ σερφάρετε στο ίντερνετ και ταυτόχρονα απαντάτε σε ένα SMS χωρίς να κλείσετε καμία εφαρμογή ή να χάσετε τη σελίδα που επισκεφτήκατε.

Το web browsing στο Android είναι ταχύτατο, υποστηρίζεται από το Adobe Flash Player και υπάρχουν πολλοί browsers για να καλύψουν και τους πλέον απαιτητικούς. Ανεξάρτητα από το κόστος, όλες οι συσκευές Android διαθέτουν GPS και Wi-fi, δικαιώνοντας έτσι το βασικό λόγο δημιουργίας του εν λόγω λειτουργικού συστήματος που δεν είναι άλλος παρά η ανεμπόδιστη και εύκολη πρόσβαση στο διαδίκτυο, σε συνδυασμό με ένα πλήθος εφαρμογών, όπως χάρτες, αναζήτηση, chat και e-mail, που πραγματικά επιτρέπουν στο χρήστη να μένει διαρκώς δικτυωμένος και ενημερωμένος.

Βασικό χαρακτηριστικό του Android, επίσης, είναι η πληθώρα εφαρμογών που υπάρχουν στο Play Store δωρεάν και μη, όπου μπορεί ο χρήστης να κατεβάσει. Ακόμη, το Android σας δίνει τη δυνατότητα να προσθέσετε widgets, δηλαδή

εικονίδια για την ταχύτερη πρόσβαση στα προγράμματα, τα οποία τοποθετούνται στη home screen του κινητού. Επιπλέον η notification bar είναι εξαιρετικά χρήσιμη, καθώς με ένα απλό drag - κατέβασμα βλέπετε όλες τις ειδοποιήσεις για τη συσκευή σας, μπορείτε να ανοιγοκλείσετε απευθείας τις υπηρεσίες χωρίς να χρειάζεται να πάτε στις ρυθμίσεις ή να προσθέσετε και άλλες λειτουργίες αναλόγως την συσκευή σας.

Όσον αφορά το hardware οι διπύρνηνοι, τετραπύρνηνοι ή και οκταπύρνηνοι πλέον επεξεργαστές και οι διακεκριμένες GPU είναι πλέον γεγονός κάνοντας την εμπειρία χρήσης όλο και καλύτερη.

### 1.3 Ιστορική Αναδρομή

Η **Android Inc.** ιδρύθηκε από τέσσερα άτομα, τους Andy Rubin, Rich Miner, Nick Sears και Chris White, ενώ η ονομασία προήλθε από το παρατσούκλι του Rubin, που ήταν βέβαια “Android”, λόγω της εμμονής του με τα ρομπότ. Αρχικά το Android στόχευε να γίνει μια πλατφόρμα για ψηφιακές κάμερες, των οποίων η αγορά τότε έδειχνε πως θα ανθίσει σύντομα, όπως και έγινε. Ωστόσο, πολύ σύντομα η προσοχή των ιδρυτών στράφηκε προς τα smartphones, που τότε κατείχαν πολύ μικρό μερίδιο της αγοράς τηλεφώνων, θεωρώντας πως δεν τους αρκούσε η αγορά των ψηφιακών μηχανών.

Αποδείχθηκε σοφή επιλογή, αφού το 2005 η Google ενδιαφέρθηκε και αγόρασε την startup. Σοφή ήταν η επιλογή και από μεριάς της Google, καθώς μέσω του Android ενισχύθηκε σημαντικά η κυριαρχία της σε κάθε επιχείρηση και σε κάθε σπίτι, πάνω σε κάθε χρήστη. Ωστόσο, από το 2005 πέρασαν 2 χρόνια για να αρχίσει η ανάπτυξη του λειτουργικού και ακόμη 1 χρόνος για να κυκλοφορήσει το Android 1.0 OS, το Σεπτέμβριο του 2008. Το **T-Mobile G1** ήταν το πρώτο Android smartphone, το οποίο πωλείτο και ως **HTC Dream** σε πολλές αγορές.



Εικόνα 2. Το πρώτο Android smartphone T-Mobile G1

Το 2009 εμφανίστηκε το **Motorola Droid** που πλασαρίστηκε στην αγορά σαν εναλλακτική λύση απέναντι στο iPhone, πουλώντας μισό εκατομμύριο συσκευές στον πρώτο μήνα κυκλοφορίας.



Εικόνα 3. Motorola Droid

Στις 28 Αύγουστου 2008 η Google ανακοινώσε το Android Market ( όπου από 6 Μάρτιου 2012 ονομάζεται Google Play ) και σταδιακά άρχισε να υποστηρίζει εφαρμογές επί πληρωμή σε Οι Android συσκευές άρχιζαν να γίνονται γνωστές και να αρέσουν στο κοινό, με αποτέλεσμα όλο και περισσότεροι διαφορές χώρες ξεχωριστά. κατασκευαστές να στραφούν προς την πλατφόρμα της Google.



Εικόνα 4. Android Market

## 1.4 Εκδόσεις Android

Όπως αναφέραμε παραπάνω το Σεπτέμβριο του 2008, τρία δηλαδή χρόνια μετά την πρώτη του παρουσίαση, κυκλοφόρησε το Android **1.0 OS** με το T-Mobile G1 που ήταν το πρώτο Android smartphone.

Πρώτο μεγάλο update ήταν η έκδοση **1.5 “Cupcake”**, τον Απρίλιο του 2009, φέρνοντας τα widgets και τους φακέλους στην homescreen. Και η έκδοση **1.6 “Donut”** δεν άργησε να την αντικαταστήσει, μόλις 5 μήνες αργότερα, υποστηρίζοντας οθόνες μεγαλύτερων αναλύσεων και με αρκετές βελτιώσεις. Βελτιώσεις που συνεχίστηκαν με ραγδαίο ρυθμό μέσω της επόμενης έκδοσης, με κωδική ονομασία **“Eclair”**, ένα μήνα αργότερα! Το Android προσπαθούσε να ωριμάσει γρήγορα για να προλάβει τις εξελίξεις, στην πραγματικότητα όμως, το ίδιο δημιουργούσε ταυτόχρονα αυτές τις εξελίξεις.



Εικόνα 5. Android 1.5 Cupcake



Εικόνα 7. Android 1.6 Donat



Εικόνα 6. Widgets στην homescreen

Μέχρι που φτάνουμε στο καίριο έτος 2010, το οποίο σημάδεψαν δύο μεγάλα updates **2.2 “Froyo”** τον Ιούνιο και **2.3 “Gingerbread”** στην εκπονή του χρόνου, το Δεκέμβριο και η κυκλοφορία του **Google Nexus One**. Η συσκευή, αν και δεν αποτέλεσε εμπορική επιτυχία, έδειξε το δρόμο για το μέλλον και προσέφερε πολύτιμη εμπειρία στους κατασκευαστές.



Εικόνα 8. Android 2.2 Froyo



Εικόνα 10. Android 2.3 Gingerbread



Εικόνα 9. Google Nexus One

Απόδειξη ήταν η αύξηση του μεριδίου της αγοράς του Android στις ΗΠΑ, αφού ξεπέρασε μέσα στο 2010 το **iOS**, χάρη κυρίως στο **πρώτο Galaxy S** της Samsung και το **Evo 4G** της HTC.



Εικόνα 11. Διαμάχη μεταξύ Android και iOS

Από εκεί και πέρα, η γιγάντωση και η εξάπλωση του Android γίνονται με τρομακτικούς ρυθμούς. Το Φεβρουάριο του 2011 κάνει την εμφάνισή της η **έκδοση 3.0 “Honeycomb”**, με κύριο στόχο τα ανερχόμενα **tablets**, η οποία γρήγορα όμως έδωσε τη θέση της στην πολύ επιτυχημένη και δημοφιλή έκδοση **4.0**, γνωστή και ως **“Ice Cream Sandwich”**.



Εικόνα 12. Android 3.0 HoneyComb



Εικόνα 13. Android 4.0 Ice Cream Sandwich

Βασιζόμενη πάνω στην Honeycomb αλλά με βελτιώσεις στα σημεία, κυριάρχησε εύκολα για ολόκληρο τον επόμενο χρόνο. Έτσι, η **4.1 “Jelly Bean”** αναπτύχθηκε πιά άνετα και αβίαστα, για να κυκλοφορήσει παράλληλα με το best seller της Samsung, το **Galaxy S III** και μετά από λίγο ακολούθησε και η **4.2** με την ίδια ονομασία. Προηγουμένως το Android είχε ξεπεράσει και τη **Blackberry** σε μερίδιο αγοράς στις ΗΠΑ, δείχνοντας πως ήρθε για να μείνει. Και τώρα έχουμε φτάσει στην έκδοση **4.4 “Kit Kat”** που διατηρεί σταθερά στην κορυφή το πολυαγαπημένο λειτουργικό.



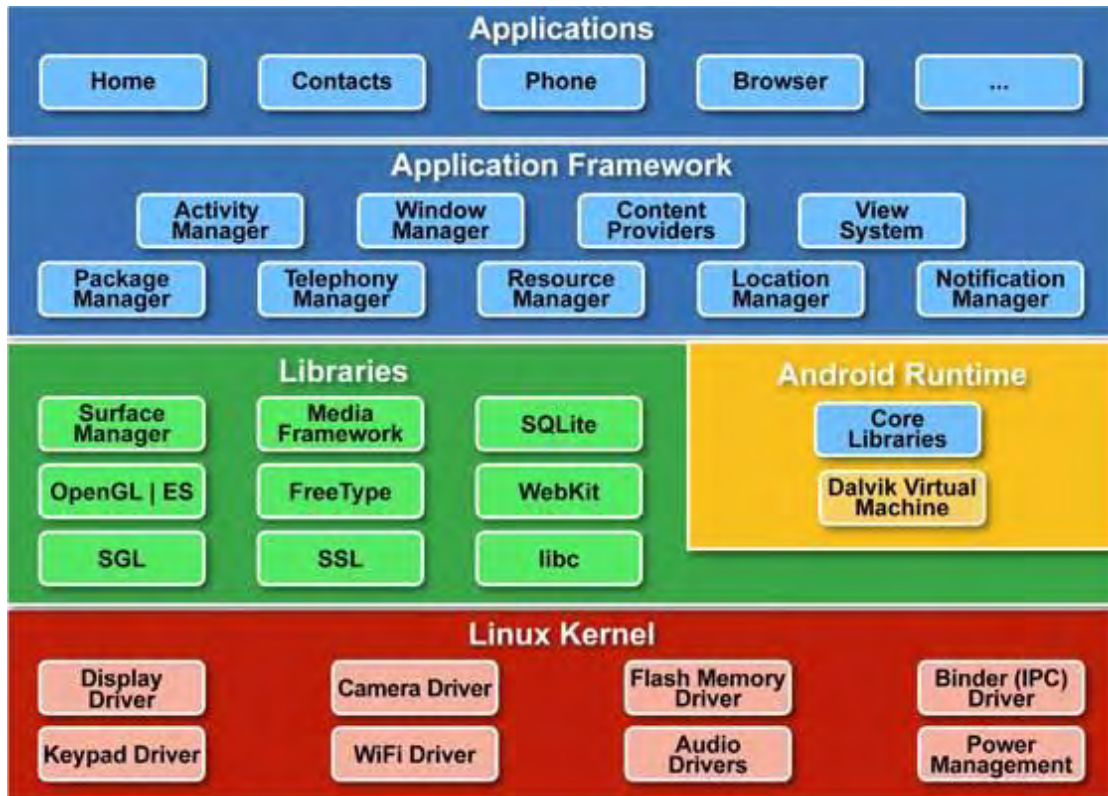
Εικόνα 15. Android 4.2 Jelly bean



Εικόνα 14. Android 4.4 Kit Kat

## 1.5 Αρχιτεκτονική Android

Το λειτουργικό σύστημα Android είναι μια στοίβα στοιχείων λογισμικού που σε γενικές γραμμές χωρίζονται σε πέντε ενότητες και τέσσερα κύρια στρώματα, όπως φαίνεται παρακάτω στο διάγραμμα της αρχιτεκτονικής.



Εικόνα 16. Διάγραμμα αρχιτεκτονικής του Android

Ο πυρήνας (**kernel**) Linux είναι το χαμηλότερο επίπεδο του συστήματος και είναι υπεύθυνο για την αλληλεπίδραση με το hardware. Επίσης προσφέρει την βασική διεργασία στο σύστημα όπως επεξεργασία δεδομένων, διαχείριση της μνήμης και διαχείριση των συσκευών – περιφερειακών όπως την όπως την κάμερα, το πληκτρολόγιο, την οθόνη κλπ. Τέλος ο πυρήνας των Linux ασχολείται και με την διαχείριση σε ότι σχέση με το δίκτυο καθώς προσφέρει και όλους τους απαραίτητους drivers που απαιτούνται.

Οι **βιβλιοθήκες (libraries)** του συστήματος βρίσκονται όπως φαίνεται στο σχεδιάγραμμα πάνω από τον πυρήνα. Εκεί περιλαμβάνονται και οι πολύ γνωστές libc, το WebKit και το SQLite Database που χρησιμοποιείται για την αποθήκευση και διαμοιρασμό δεδομένων της εφαρμογής. Συνήθως είναι προγραμματισμένες σε γλώσσα C/C++ και η πρόσβαση σε αυτές δεν είναι δυνατή καθώς χρησιμοποιούνται αυτές του ανώτερου επιπέδου.

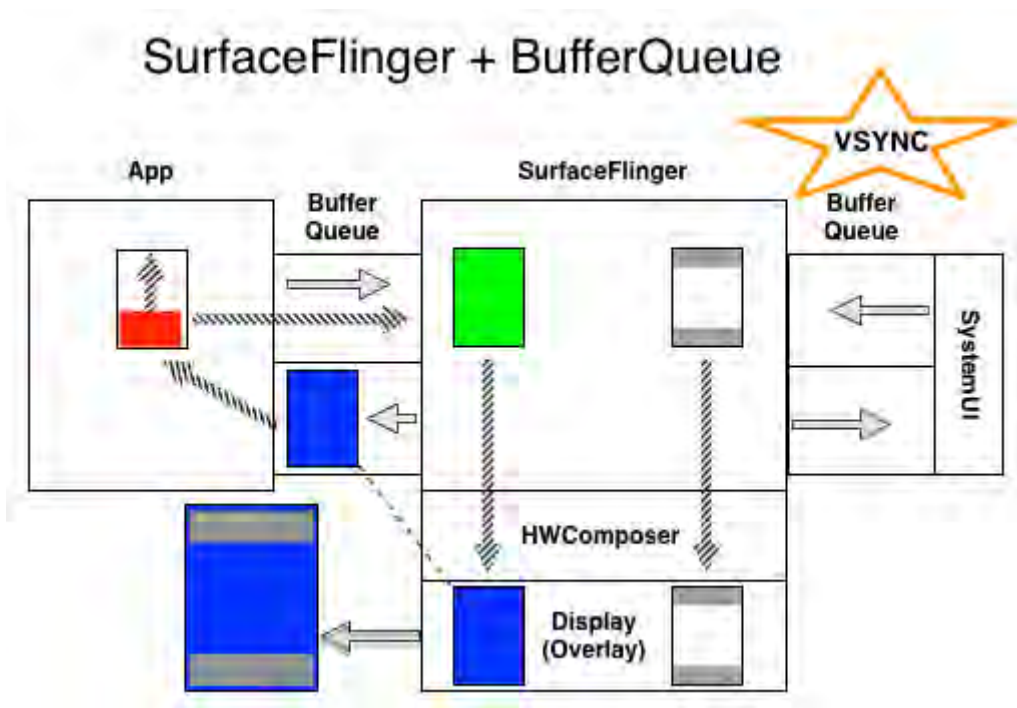
Τρίτο στη σειρά είναι το **Android Runtime** που περιλαμβάνει ένα πολύ σημαντικό στοιχείο το **Dalvik Virtual Machine** που είναι σαν ένα είδος Java εικονικής μηχανής ειδικά σχεδιασμένη και βελτιστοποιημένη για Android. Το



τελευταίο χρησιμοποιεί τα χαρακτηριστικά του πυρήνα του Linux όπως την διαχείριση της μνήμης και το multi-threading που είναι εγγενή για την γλώσσα προγραμματισμού Java. Ακόμη αυτό που το κάνει ξεχωριστό είναι ο σχεδιασμός του, που δίνει την δυνατότητα σε κάθε εφαρμογή να τρέχει σε δικό της εικονικό μηχανήμα. Έτσι βελτιώνεται και η ασφάλεια και η ταχύτητα στο λειτουργικό.

Το **πλαίσιο εφαρμογών (Application Framework)** περιέχει ένα πλαίσιο ανάπτυξης που τυποποιεί και διευκολύνει τον προγραμματισμό των Android εφαρμογών. Αυτές οι βιβλιοθήκες είναι γραμμένες σε Java και μπορούν να χρησιμοποιηθούν από τους developers. Κύριο στοιχείο του είναι η επαναχρησιμοποίηση των αρχείων κάθε εφαρμογής καθώς μπορούν να δημοσιεύσουν τις δυνατότητές τους και να χρησιμοποιηθούν από κάποια άλλη στο μέλλον.

Οι **εφαρμογές (applications)** είναι το βασικό στοιχείο κάθε λειτουργικού και το Android από προεπιλογή διαθέτει μια γκάμα προεγκατεστημένων εφαρμογών όπως το Ημερολόγιο, Επαφές, Μηνύματα SMS κλπ. Μέσα από το Google Play Store και άλλα markets που κυκλοφορούν ο χρήστης όμως μπορεί να εγκαταστήσει όποια εφαρμογή θέλει.



Εικόνα 17. Λειτουργία του Android

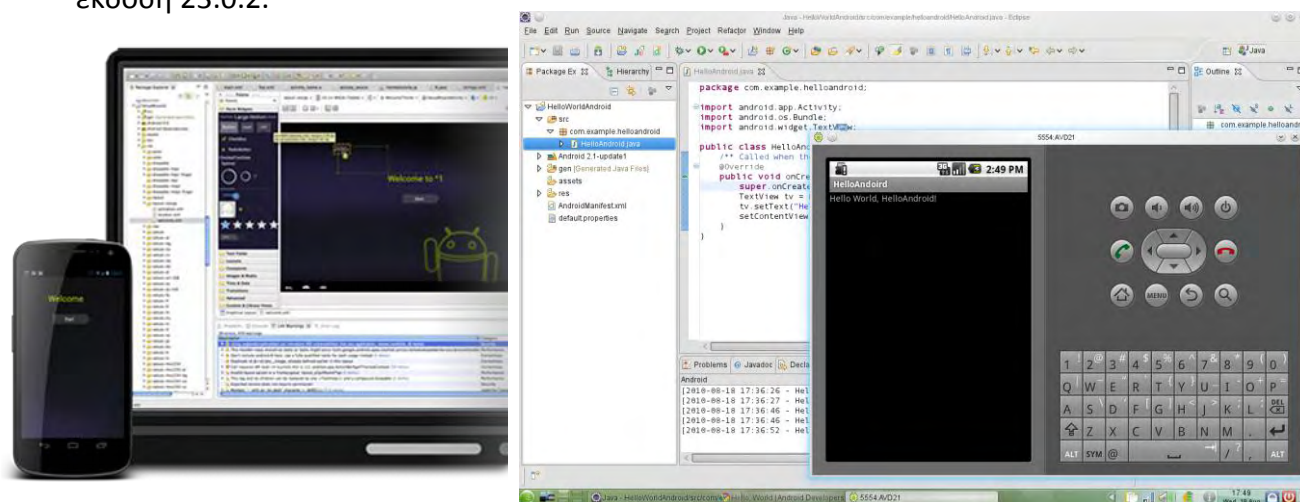
## 2. Προγραμματισμός στο Android

### 2.1 Android SDK

Το Android SDK παρέχει στον developer τις απαραίτητες API libraries και τα εργαλεία που χρειάζεται για να αναπτύξει μόνος του και να τεστάρει την εφαρμογή του. Πιο συγκεκριμένα περιλαμβάνει τον debugger και διάφορες βιβλιοθήκες για τον χειρισμό των διάφορων αισθητήρων του smartphone – tablet. Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η Java αλλά η Google δίνει την δυνατότητα στον developer μέσω του NDK (Native Development Kit) να ενσωματώσει διάφορα κομμάτια της εφαρμογής του που είναι σε γλώσσα C/C++. Επίσης πρόσφατα η Google έχει λανσάρει και το Eclipse ADT (Android Developer Tools) Bundle το οποίο αποτελεί μια εύκολη λύση για τους νέους developers καθώς περιλαμβάνει το Eclipse με προεγκατεστημένες τις απαραίτητες βιβλιοθήκες και τον debugger έτοιμο προς χρήση.

Το SDK χρειάζεται κάποιο IDE (Integrated Development Environment) περιβάλλον στο οποίο πρέπει να εγκατασταθεί για να λειτουργήσει όπως π.χ. το Eclipse. Σε κάθε περίπτωση ο προγραμματιστής πρέπει να επιλέξει αυτός τα build tools και τον debugger για την έκδοση του Android που στοχεύει να είναι η εφαρμογή του. Σωστό πάντως θα ήταν όπως αναφέρει και η Google να επιλέγεται πάντα ως στόχος η τελευταία έκδοση του Android που χρησιμοποιούν όλες σχεδόν οι καινούριες συσκευές.

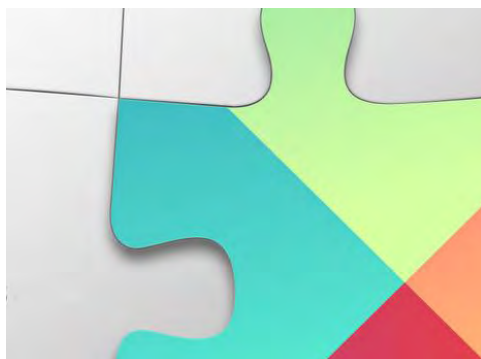
Οι πλατφόρμες στις οποίες υποστηρίζεται το SDK είναι οι τρεις πιο δημοφιλείς, δηλαδή τα Windows XP/7, MAC OS X 10.5.8 και μετά καθώς και κάθε σύγχρονη διανομή Linux. Το SDK αυτή την στιγμή (Αύγουστος 2014) βρίσκεται στη έκδοση 23.0.2.



Εικόνα 18. Eclipse με εγκατεστημένο το Android SDK

## 2.2 Google Play Services

Οι υπηρεσίες Google Play προσθέτουν στην εφαρμογή πολλά χαρακτηριστικά που την κάνουν ελκυστική για κάθε χρήστη. Με αυτές ο developer μπορεί να ενσωματώσει στην εφαρμογή του διάφορες υπηρεσίες όπως το Google Maps, Google+ και πολλά άλλα με την δυνατότητα αυτόματης αναβάθμισης από το Play Store προσφέροντας στον χρήστη όλα τα καλά της Google.



Εικόνα 19. Google Play Services

Για να χρησιμοποιηθούν αυτές οι υπηρεσίες στην εφαρμογή πρέπει να εγκατασταθεί το Google Services SDK μέσω της επιλογής Google API της εκάστοτε έκδοσης που χρησιμοποιείται. Με τις νέες βιβλιοθήκες η εφαρμογή μπορεί να αποκτήσει εύκολα όλα τα νέα χαρακτηριστικά της Google. Τα Google Services τρέχουν στο background της συσκευής και μέσω του client library συνδέεται με την εφαρμογή και εκτελεί ό,τι χρειάζεται.

Δεν υπάρχει λόγος ανησυχίας για την συμβατότητα καθώς όλες σχεδόν οι καινούριες συσκευές με Android 2.3 και πάνω τις υποστηρίζουν.

### 2.2.1 Google Maps

Μια υπηρεσία της Google ευρέως γνωστή είναι οι χάρτες της Google Maps. Με το Google Maps Android API v2 ο developer μπορεί να χρησιμοποιήσει τους χάρτες στην εφαρμογή του.

Για να γίνει αυτό πρέπει πρώτα να πάρει ένα κλειδί από την Google για να του επιτρέψει να τους χρησιμοποιήσει. Μέσα από το **Google APIs Console** όπου ο developer πρέπει να έχει κάνει login με το google account του, στην καρτέλα **Services** πρέπει να ενεργοποιήσει το Google Maps Android API v2. Στην συνέχεια

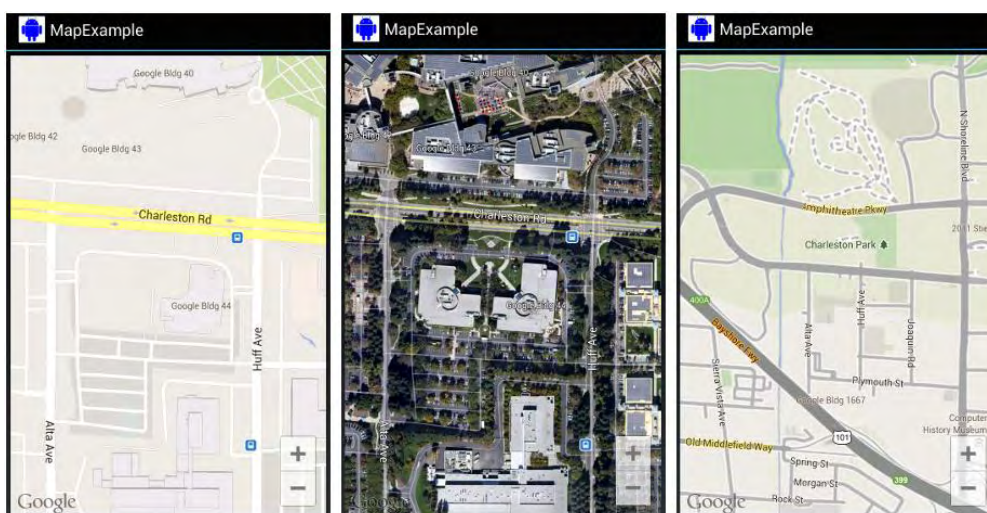
στο **API Access** να πατήσει **Create New Android Key** να συμπληρώσει το SHA-1 fingerprint που θα βρει στο Eclipse στις Ιδιότητες -> Debug του project του και το package name. Τελικά θα του εμφανιστεί το κλειδί που χρειάζεται να βάλει στο manifest αρχείο του project για να λειτουργήσουν οι χάρτες. Φυσικά πρέπει να συμπεριληφθούν σε αυτό και οι απαραίτητες άδειες για την χρήση του ιντερνέτ και της τρέχουσας θέσης GPS:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```



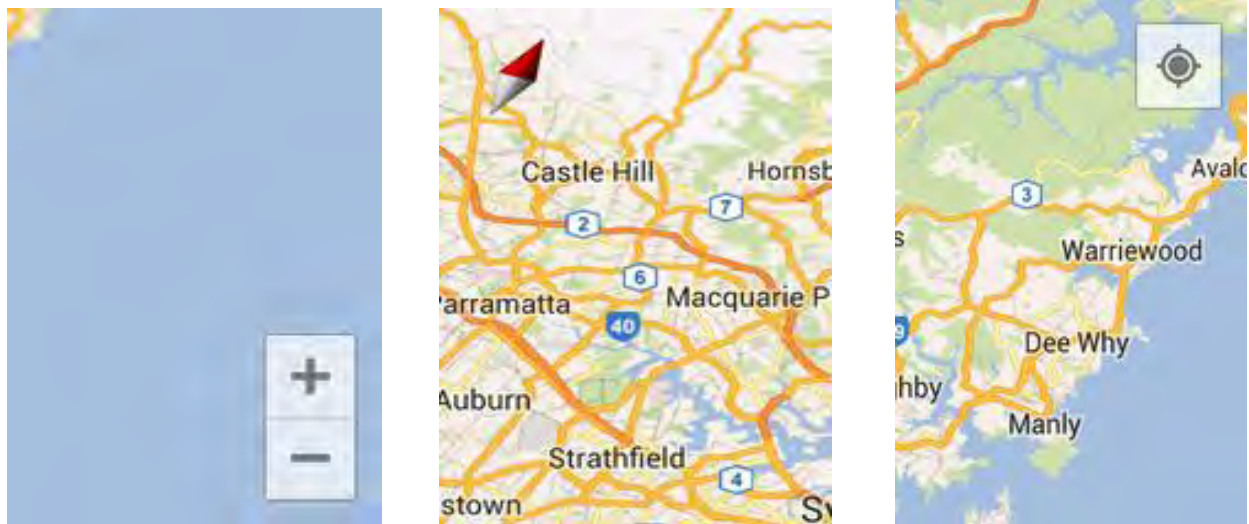
Εικόνα 20. Google Maps

Εφόσον οι χάρτες έχουν εισαχθεί ο developer μπορεί να τους χρησιμοποιήσει όπως εκείνος θέλει. Μερικές από τις δυνατότητες που δίνουν οι χάρτες της Google στους developers είναι να ενεργοποιήσουν την δυνατότητα να μπορεί ο χρήστης να επιλέγει τον τύπο του χάρτη που τον βολεύει ανάμεσα σε Κανονικό, Δορυφόρο και Υβριδικό.



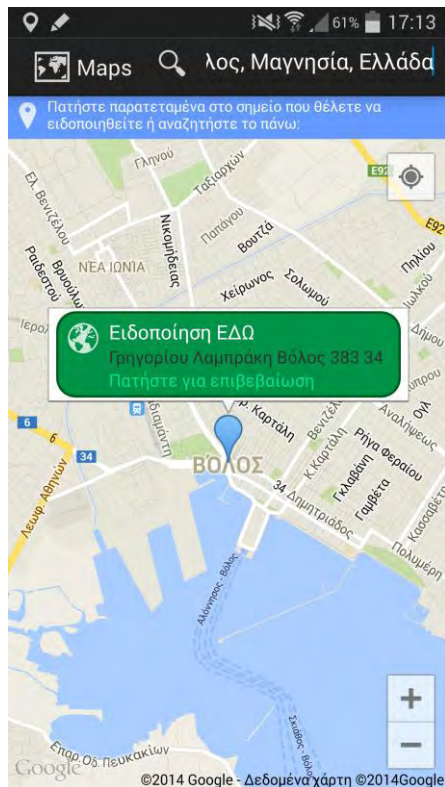
Εικόνα 21. Κανονική, Δορυφορική και Υβριδική (Terrain) εμφάνιση Google Maps

Επίσης κάποιες άλλες δυνατότητες που προσφέρουν οι χάρτες και μπορεί να προσθέσει ο developer είναι τα **gesture controls** με τα οποία ο χρήστης μπορεί να περιστρέφει τον χάρτη να κάνει zoom κ.α. , την **compass** που εμφανίζεται σαν κουμπί στον χάρτη και αν πατηθεί περιστρέφει τον χάρτη έτσι ώστε να δείχνει πάντα τον βορρά και το **my location button** το οποίο αν πατηθεί εμφανίζει στον χάρτη την τρέχουσα θέση του χρήστη.

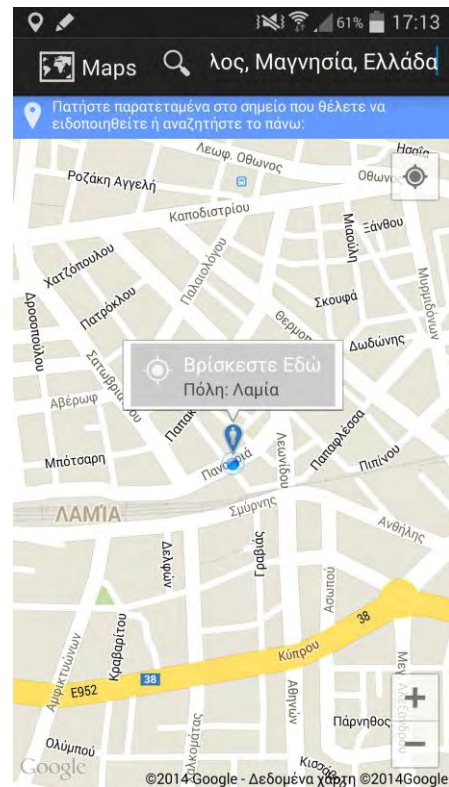


Εικόνα 22. Zoom Controls, Compass και My Location Button

Τέλος ένα άλλο σημαντικό feature που έχουν οι χάρτες είναι η δυνατότητα να αλληλεπιδρά ο χρήστης με αυτούς τοποθετώντας markers (Πινέζες πάνω στον χάρτη) και info windows (μικρά παραθυράκια με πληροφορίες πάνω από την πινέζα). Ο developer μπορεί να επεξεργαστεί την εμφάνιση αυτών και να ενεργοποιήσει πλήθος ενεργειών που προσφέρουν. Για παράδειγμα την παρούσα εφαρμογή στους χάρτες που έχω χρησιμοποιήσει, έχω ενεργοποιήσει το My Location Button όπου πατώντας εμφανίζεται ένα marker με μια εικόνα ανθρώπου και ένα info window που γράφει την τρέχουσα πόλη, τα Gesture Controls και τα markers – info windows τα οποία έχω επεξεργαστεί για να εμφανίζεται μόνο ένα marker και όχι παραπάνω αλλά και το info window του marker να είναι πράσινο γράφοντας την τρέχουσα διεύθυνση με μαύρα γράμματα. Όλα αυτά φαίνονται στις δύο παρακάτω φωτογραφίες που είναι από την εφαρμογή.



**Εικόνα 24.** Marker και InfoWindow για μια περιοχή ειδοποίησης στον Βόλο



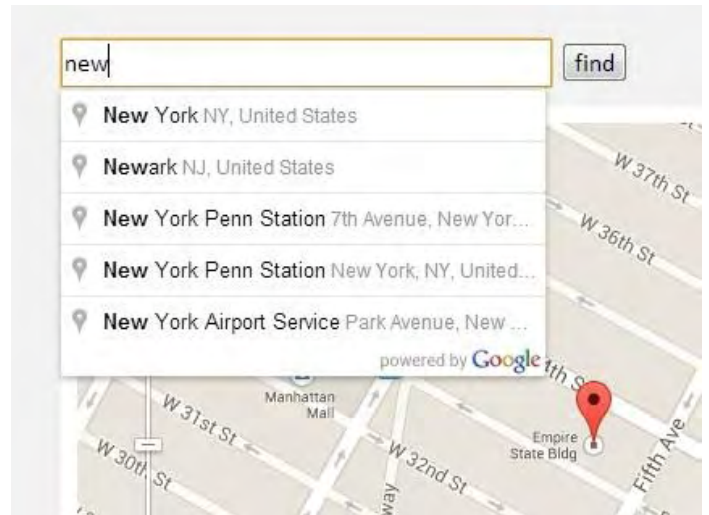
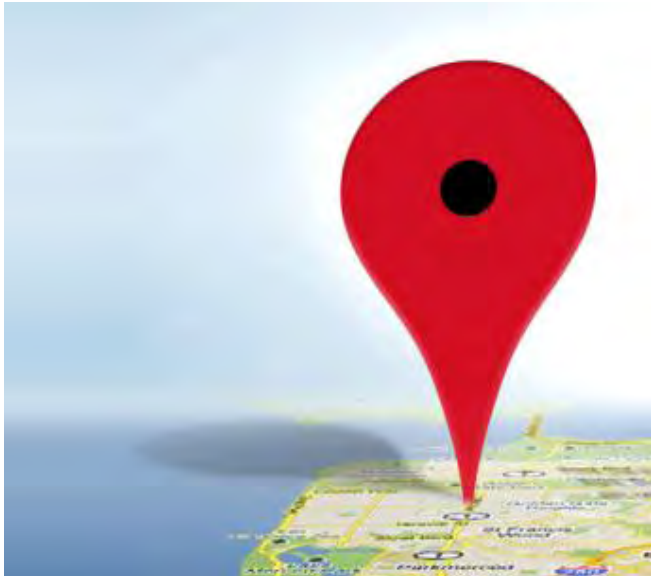
**Εικόνα 23.** Πατώντας το My Location Button εμφάνιση Marker και στο InfoWindow η πόλη

## 2.2.2 Google Places

Μια άλλη υπηρεσία της Google που χρησιμοποιείται ευρέως και αυτή μαζί με τους χάρτες είναι το Google Places που χρησιμοποιείται για την γραπτή αναζήτηση διευθύνσεων ή και τοποθεσιών στον χάρτη. Ο developer μπορεί να χρησιμοποιήσει το Google Places API για την εφαρμογή του. Όπως και στους χάρτες έτσι και εδώ πρέπει να ενεργοποιήσει το Places από το Google API Console και να δημιουργήσει ένα νέο key για browser applications μιας και το Places API είναι σχεδιασμένο για χρήση στο διαδίκτυο.

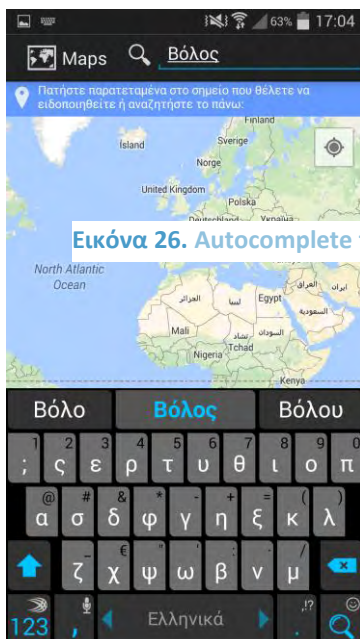
Ο κύριος λόγος που χρησιμοποιείται είναι το γνωστό auto complete που χρησιμοποιεί ο developer στην εφαρμογή και αφορά την αυτό-συμπλήρωση διευθύνσεων και τοποθεσιών που είναι εγγεγραμμένη στην υπηρεσία. Στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ως εναλλακτική στη αναζήτηση της τοποθεσίας ειδοποίησης και με την επιλογή της διεύθυνσης που πληκτρολογήθηκε από τον χρήστη ένα marker, που αναφέραμε στους χάρτες, τοποθετείται σε εκείνο το μέρος δηλώνοντάς το για την ειδοποίηση του χρήστη. Η χρήση του Places API όπως αναφέραμε χρησιμοποιεί αντικείμενα JSON, γλώσσας που χρησιμοποιείται

στο web development για επικοινωνία με τον server και την επιστροφή των αποτελεσμάτων.



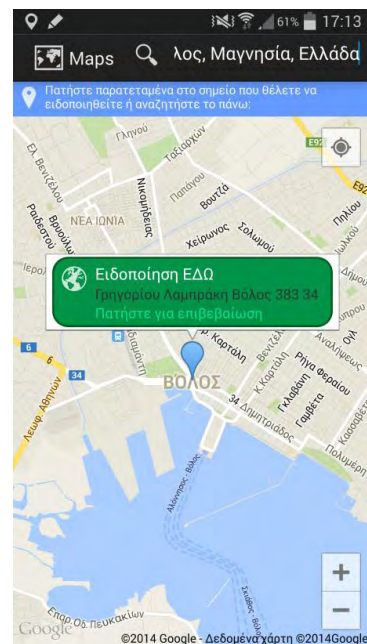
Εικόνα 25. Υπηρεσία Google Places

Στην παρακάτω φωτογραφία φαίνεται η εφαρμογή μόλις έχει πατηθεί στο auto complete το μέρος της ειδοποίησης που θέλει ο χρήστης και η αυτόματη τοποθέτηση marker σε εκείνη την τοποθεσία στον χάρτη.



Εικόνα 26. Autocomplete τοποθεσίας στον χάρτη

Εικόνα 27. Αναζήτηση της τοποθεσίας στο autocomplete με Google Places

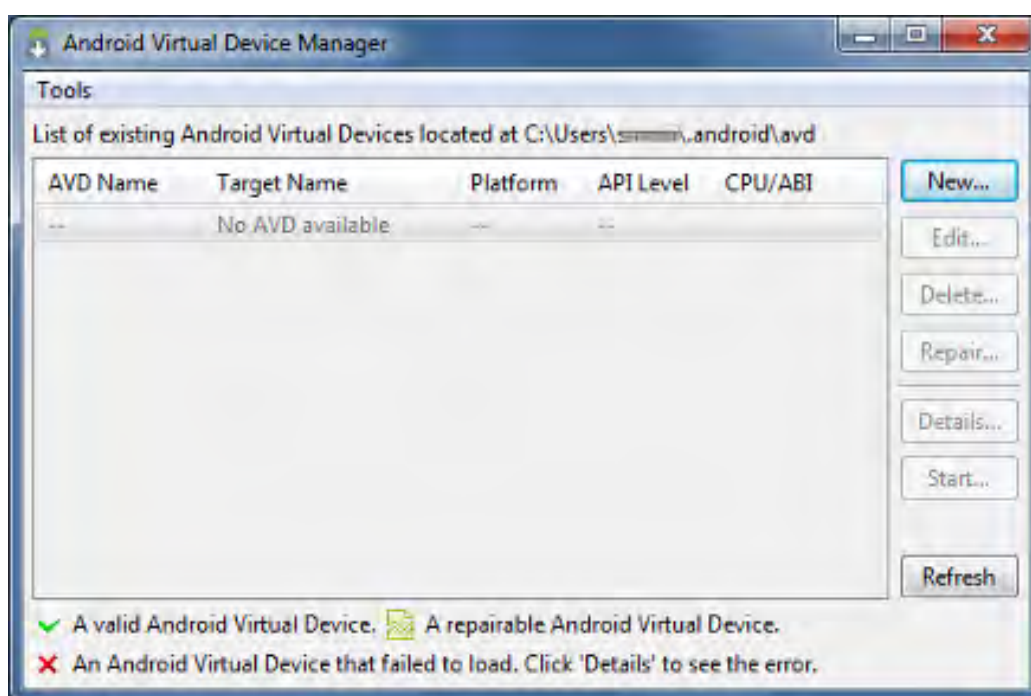


Εικόνα 28. Τοποθέτηση πινέζας στην επιλεγμένη τοποθεσία

## 2.3 Android Emulator

Το Android SDK περιλαμβάνει ένα κινητό τηλέφωνο εξομοιωτή, δηλαδή μια virtual κινητή συσκευή που τρέχει στον υπολογιστή του developer χωρίς να χρειάζεται η χρήση φυσικής συσκευής. Σκοπός της είναι να μπορεί ο developer να δοκιμάζει την εφαρμογή του χωρίς την χρήση κάποιας εξωτερικής συσκευής αλλά με όλες τις δυνατότητες που αυτή προσφέρει.

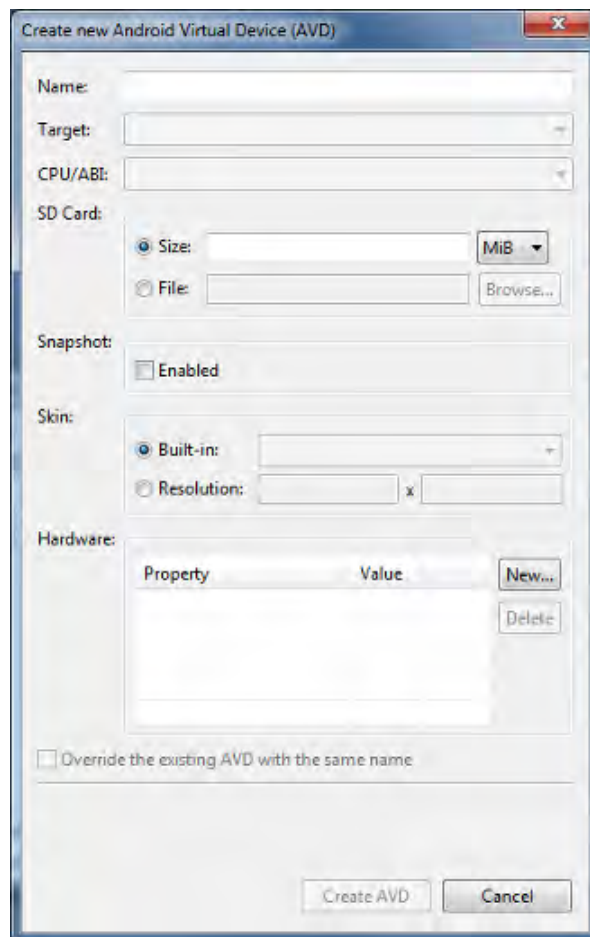
Για να την χρησιμοποιήσει ο developer πρέπει μέσα από το Eclipse να ανοίξει το AVD (Android Virtual Device) Manager και πατώντας το κουμπί New να προσθέσει μια νέα συσκευή.



Εικόνα 29. AVD Manager στο Eclipse

Μόλις πατήσει το κουμπί θα του εμφανιστεί ένα νέο παράθυρο όπου θα ζητάει στον χρήστη να συμπληρώσει τα στοιχεία της νέας virtual συσκευής.





Εικόνα 30. Προσθήκη νέας συσκευής (Emulator) στο Eclipse

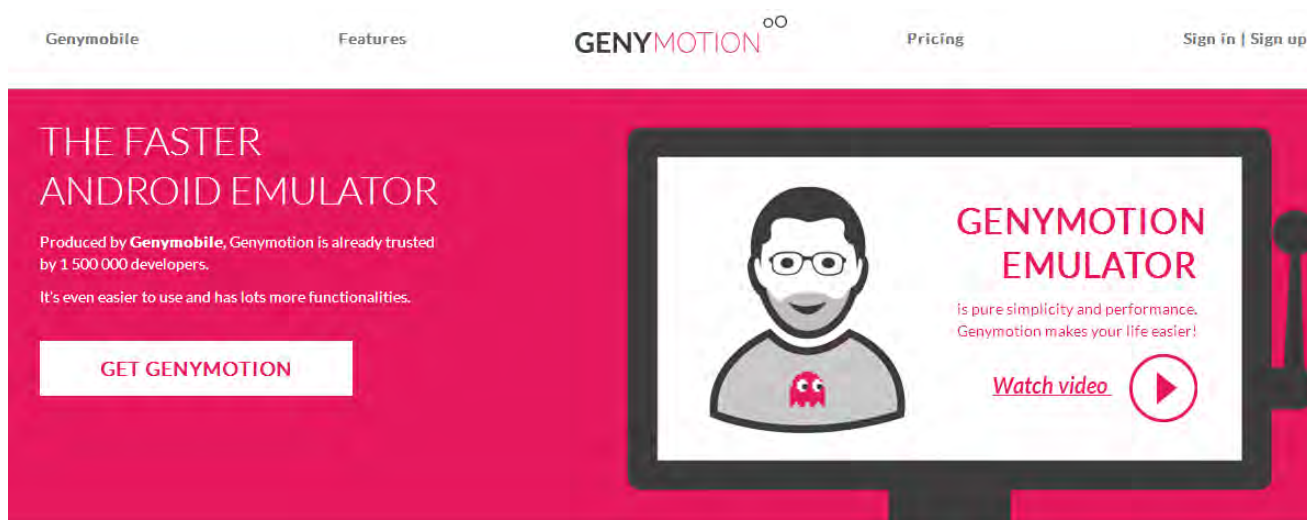
Σε αυτό το νέο παράθυρο ο developer πρέπει να πληκτρολογήσει τα στοιχεία της νέας του συσκευής όπως το όνομα, το μέγεθος αποθήκευσης, το μέγεθος της RAM, την εμφάνισή του καθώς και αν το target του είναι Android API ή Google API ανάλογα με το είδος της εφαρμογής. Το τελικό emulator θα εμφανίζεται όπως φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 31. Android Emulator

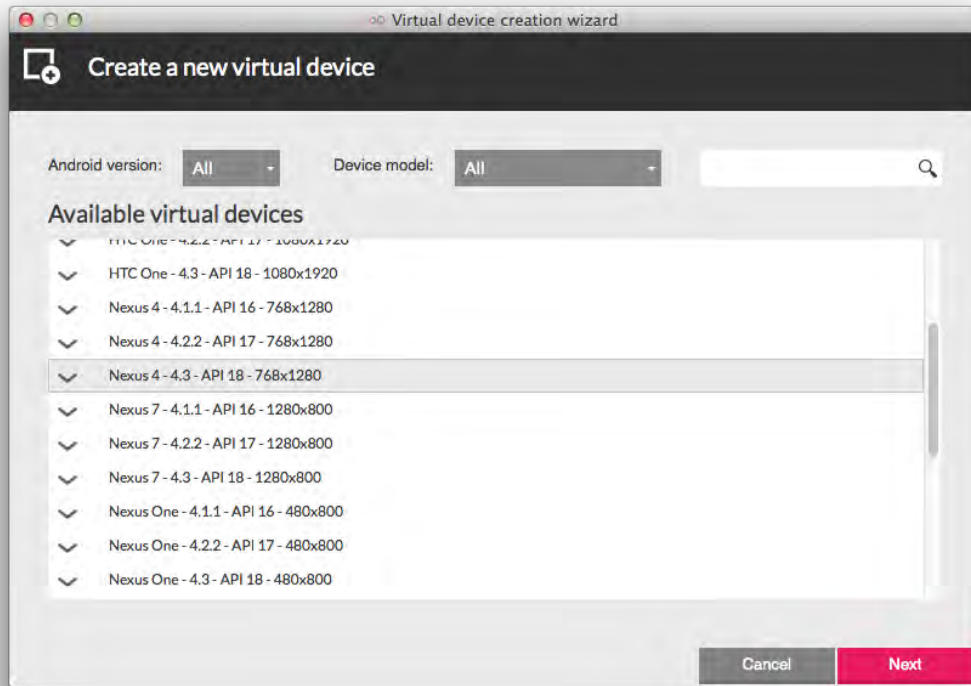
### 2.3.1 Gevymotion Emulator

Το Gevymotion είναι ένα ολοκληρωμένο σετ εργαλείων που προσφέρουν στον developer ένα εικονικό περιβάλλον με Android. Είναι διαθέσιμο για τα πιο γνωστά λειτουργικά Windows, Linux και Mac OSX. Η εγκατάσταση και η χρήση του είναι πολύ απλή αφού το μόνο που χρειάζεται είναι μια εγκατάσταση και τίποτα άλλο. Τα αρχεία που χρειάζεται τα φορτώνει από το ιντερνέτ γι αυτό και είναι απαραίτητο ο χρήστης να είναι συνδεδεμένος σε αυτό για να το χρησιμοποιήσει.



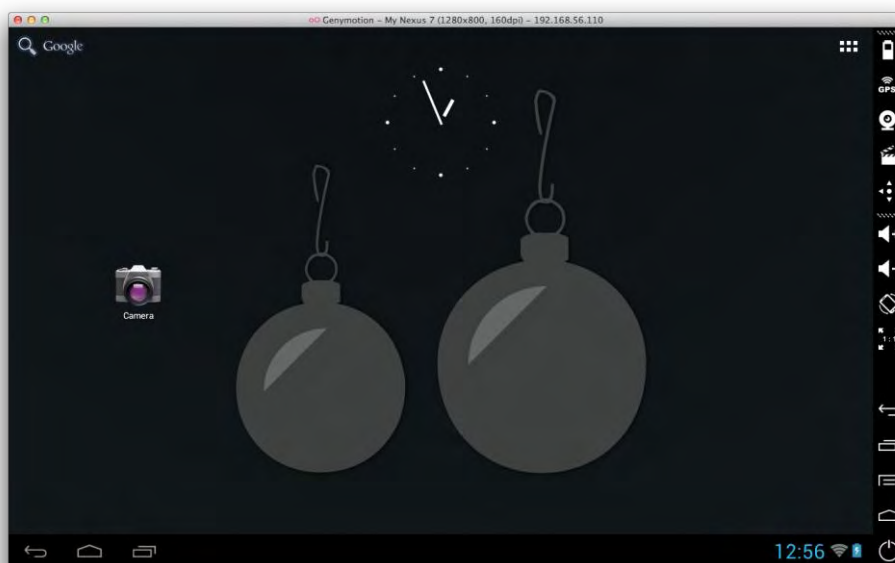
Εικόνα 32. Ιστοσελίδα της Gevymotion

Όταν τελειώσει η εγκατάσταση του προγράμματος ο χρήστης επιλέγει το κινητό τηλέφωνο που θέλει να χρησιμοποιήσει σαν emulator και την έκδοση του Android μέσα από μια μεγάλη γκάμα συσκευών και εκδόσεων. Το Eclipse αναγνωρίζει κάθε φορά την συσκευή και τεστάρει – τρέχει την εφαρμογή εκεί. Το συγκεκριμένο emulator υστερεί όμως των υπηρεσιών Play Services όπως και ο standard emulator του Android όμως μέσα από έναν οδηγό που βρίσκεται στο ιντερνέτ η εγκατάστασή τους είναι εφικτή.



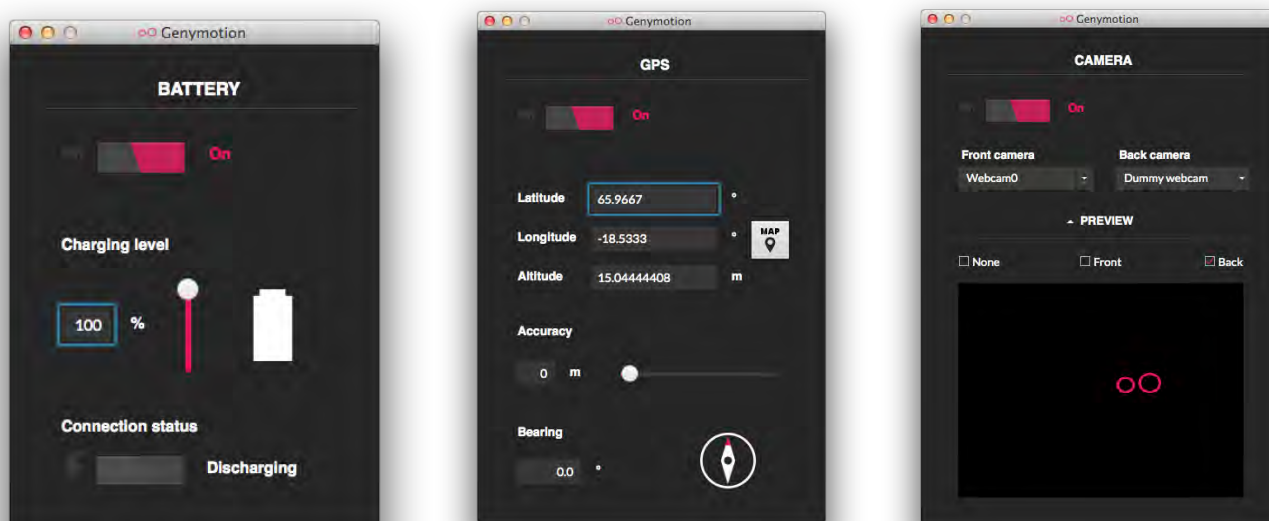
Εικόνα 33. Επιλογή συσκευής και έκδοσης Genymotion Emulator

Αυτό που κάνει τον συγκεκριμένο εξομοιωτή να ξεχωρίζει είναι η ικανότητα να πειράζει ο developer, εικονικά πάντα, τα περιφερειακά της συσκευής όπως την θέση του GPS, το ποσοστό της μπαταρίας καθώς και την κάμερα για να μπορέσει να τεστάρει την εφαρμογή του σε virtual περιβάλλον και όχι σε πραγματική συσκευή.



Εικόνα 34. Nexus 7 Emulator

Αυτό του δίνει το πλεονέκτημα να παρακολουθεί την εκτέλεση της εφαρμογής σε κατάσταση που έχει δημιουργήσει κάνοντας το τεστάρισμα μια εύκολη δουλειά. Επίσης μπορεί να χρησιμοποιήσει την κάμερα του υπολογιστή σαν κάμερα του κινητού, να ορίσει εικονικά την θέση που βρίσκεται το κινητό ανά πάσα στιγμή μέσα από τον παγκόσμιο χάρτη με βάση το latitude και το longitude και το ποσοστό της μπαταρίας που θα ήταν ιδιαίτερα χρήσιμο για το τεστάρισμα εφαρμογής που ασχολείται με την τελευταία.



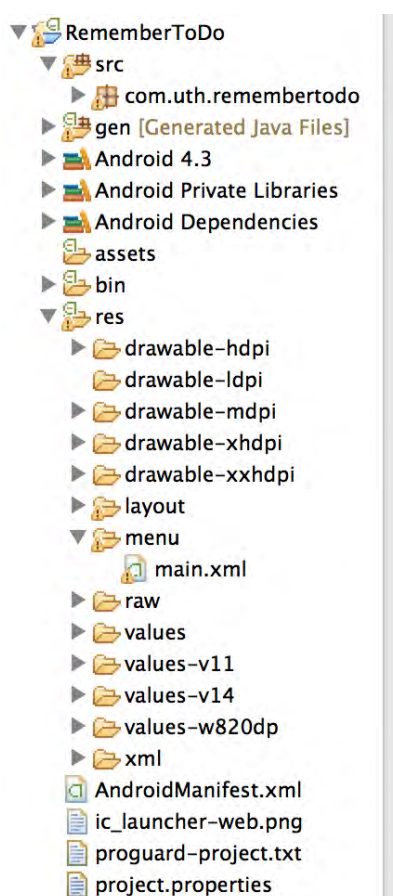
Εικόνα 35. Εικονική επεξεργασία της μπαταρίας, της θέσης στο GPS και της κάμερας από το Genymotion Emulator

Τέλος όπως αναφέραμε προηγουμένως η δυνατότητα να μπορεί να χρησιμοποιεί, ύστερα από εγκατάσταση του developer, τις υπηρεσίες Google βοηθάει πολύ τους developers που έχουν συμπεριλάβει στην εφαρμογή τους Google Maps, Google Places κλπ καθώς ο μόνος τρόπος να την ελέγξουν θα ήταν μια φυσική συσκευή με Android συνδεδεμένη με τον υπολογιστή.

## 3. Περιεχόμενα Εφαρμογής Android

### 3.1 Βασικές Πληροφορίες

Στο Android οι εφαρμογές είναι φτιαγμένες σαν ένα συνδυασμό από ανεξάρτητα στοιχεία που μπορούν να επικοινωνούν μεταξύ τους. Όλες σχεδόν αποτελούνται από Activities και τα αντίστοιχα xml αρχεία, το manifest αρχείο, εικόνες διάφορων μεγεθών, αρχεία ήχου και ότι άλλο απαιτεί η υλοποίηση της κάθε εφαρμογής. Παρακάτω επεξηγούνται αναλυτικά τα βασικά στοιχεία που περιέχει κάθε εφαρμογή και επεξηγείται ο τρόπος που αυτά συνδυάζονται.

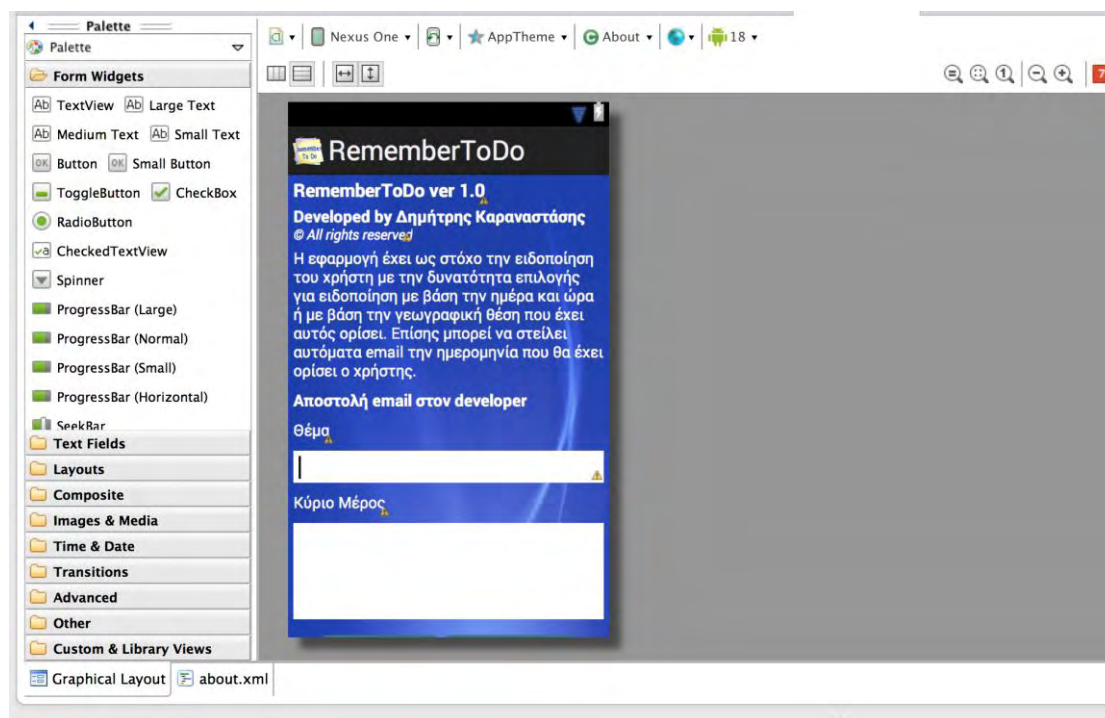


Εικόνα 36. Οι φάκελοι και υποφάκελοι της εφαρμογής που περιέχουν τα xml αρχεία και τα Activities όπως φαίνονται στο project manager του Eclipse

#### Αρχεία XML

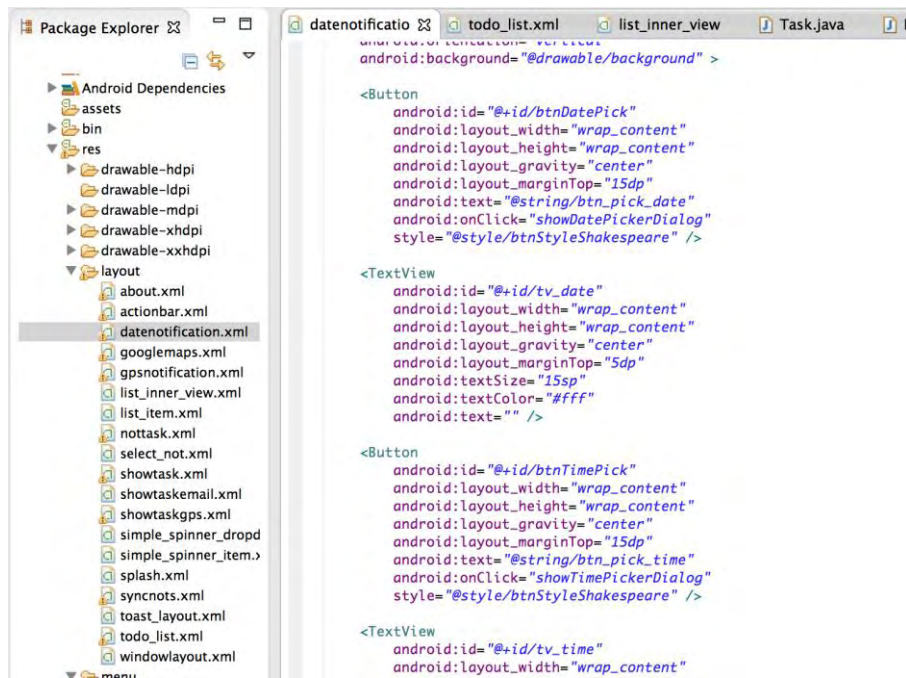
Οι εφαρμογές που ανεβαίνουν στο Play Store δουλεύουν σχεδόν σε κάθε συσκευή Android. Αυτό γίνεται χάρη στον τρόπο που δημιουργείται το γραφικό περιβάλλον καθώς κάθε διαφορετικό παράθυρο αποτελείται από ένα **xml** αρχείο

στο οποίο ο developer ορίζει ένα layout και προσθέτει ότι άλλα στοιχεία επιθυμεί δίνοντάς τους συγκεκριμένα ID's για να τα κάνει λειτουργικά αργότερα στον κώδικα. Επίσης ορίζει και το μέγεθος των στοιχείων με μονάδα μέτρησης το dp που καταλαβαίνει την ανάλυση κάθε οθόνης και προσαρμόζει αναλόγως το μέγεθος έτσι ώστε να φαίνονται το ίδιο καλά σε κάθε συσκευή και να μην αλλοιώνεται η ανάλυση. Τα στοιχεία που μπορεί ο χρήστης να προσθέσει είναι εμφανή εκ των προτέρων στο Eclipse και ο developer μπορεί να επιλέξει από μια μεγάλη γκάμα απλά επιλέγοντας το στοιχείο που θέλει να χρησιμοποιήσει και “πετώντας” το μέσα στο παράθυρο δημιουργείται και ο αντίστοιχος κώδικας στο xml αρχείο.



Εικόνα 37. Δημιουργία xml αρχείου στο Eclipse και τα διάφορα στοιχεία που μπορεί να χρησιμοποιήσει ο developer

Από εκεί μετά μπορεί να προσθέσει κώδικα σε xml android γλώσσα και να ορίσει το id, το μέγεθος, το χρώμα και όλα τα γραφικά χαρακτηριστικά του κάθε στοιχείου. Το id πρέπει να είναι μοναδικό για κάθε στοιχείο καθώς αργότερα που θα περιληφθούν στα Activities δεν θα μπορεί να τα ξεχωρίσει ο developer. Επίσης το μέγεθος κάθε στοιχείου όπως είπαμε και παραπάνω πρέπει να είναι σε dp και των γραμμάτων σε sp για να προσαρμόζεται στην αντίστοιχη ανάλυση κάθε συσκευής αυτόματα. Οι επιλογές που έχει στην γραφική επεξεργασία του κάθε στοιχείου είναι πάρα πολλές και μερικές μοναδικές για το καθένα. Τέλος πρέπει να αναφέρουμε ότι όλα τα στοιχεία περιλαμβάνονται σε ένα layout και κάθε φορά ο developer μπορεί να επιλέξει ανάλογα με τις γραφικές απαιτήσεις του κάθε xml το κατάλληλο ανάμεσα στο Frame/Linear/Grid Layout όπου το καθένα έχει διαφορετικά χαρακτηριστικά. Φυσικά το καθένα μπορεί να χρησιμοποιηθεί πολλές φορές μέσα σε ένα άλλο.



Εικόνα 38. Κώδικας xml που ορίζει τα χαρακτηριστικά κάθε στοιχείου που υπάρχει μέσα στο datenotification xml

Τα αρχεία αυτά βρίσκονται μέσα στον φάκελο layout του project και στην συνέχεια εισάγονται στο Activity για να αποκτήσουν “ζωή”. Σε περίπτωση που ο developer επιθυμεί να δημιουργήσει γραφικά και για τον οριζόντιο προσανατολισμό του εκάστοτε xml, δηλαδή για το portrait mode τότε μπορεί φτιάχνοντας ένα xml με το ίδιο όνομα που υπάρχει στο φάκελο layout και αποθηκεύοντάς το σε έναν νέο φάκελο layout-land που πρέπει να δημιουργήσει ο ίδιος.

### Αρχεία Activities

Δεν φτάνει μόνο το γραφικό περιβάλλον για να λειτουργήσει μια εφαρμογή. Για να γίνει λειτουργική υπάρχει κώδικας από πίσω σε γλώσσα Java όπως έχουμε προαναφέρει που δίνει “ζωή” στα κομμάτια του xml αρχείου και ονομάζεται Activity. Κάθε activity έχει το δικό της γραφικό περιβάλλον που δημιουργείται στο xml αρχείο και το δηλώνει χρησιμοποιώντας τον παρακάτω κώδικα.

```
public class DateAndTimeNotification extends Activity implements OnClickListener {
    |
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);

        ActionBar actionBar = getActionBar();
        actionBar.setTitle("Date - Time Notification");
        actionBar.setIcon(R.drawable.ic_action_alarms);

        setContentView(R.layout.datenotification);
    }
}
```

Όπως παρατηρούμε πρέπει να κάνουμε extends το Activity και στην συνέχεια να χρησιμοποιήσουμε την συνάρτησή της onCreate για να κάνουμε αντιστοίχιση - setContentView το xml αρχείο που της έχουμε φτιάξει. Το ActionBar αφορά την μπάρα που εμφανίζεται κάθε φορά στο πάνω μέρος της εφαρμογής και σε κάθε Activity μπορεί ο developer να ορίσει τον τίτλο και το εικονίδιο δίπλα του.

Φυσικά σε μια εφαρμογή υπάρχουν παραπάνω από μια Activities, μια για κάθε διαφορετικό παράθυρο της εφαρμογής σχεδόν. Όλες αυτές μαζί με κάθε άλλο κομμάτι κώδικα σε Java περιλαμβάνονται στο src φάκελο του project. Στο Activity αρχικά αναθέτονται σε μεταβλητές τα στοιχεία του xml αρχείου, αν υπάρχουν, για να χρησιμοποιηθούν μέσα σε αυτή. Για να γίνει αυτό χρησιμοποιούνται τα id's που αναφέραμε παραπάνω και ο τύπος του κάθε στοιχείου όπως φαίνεται στην παρακάτω εικόνα.

```
tv_date = (TextView) findViewById(R.id.tv_date);
tv_time = (TextView) findViewById(R.id.tv_time);
btn_sendDateNot = (Button) findViewById(R.id.b_sendDateNot);
btn_clearDateNot = (Button) findViewById(R.id.b_clearDateNot);
edt_title = (EditText) findViewById(R.id.edt_title);
edt_mainDateNot = (EditText) findViewById(R.id.edt_mainDateNot);

btn_sendDateNot.setOnClickListener(this);
btn_clearDateNot.setOnClickListener(this);
```

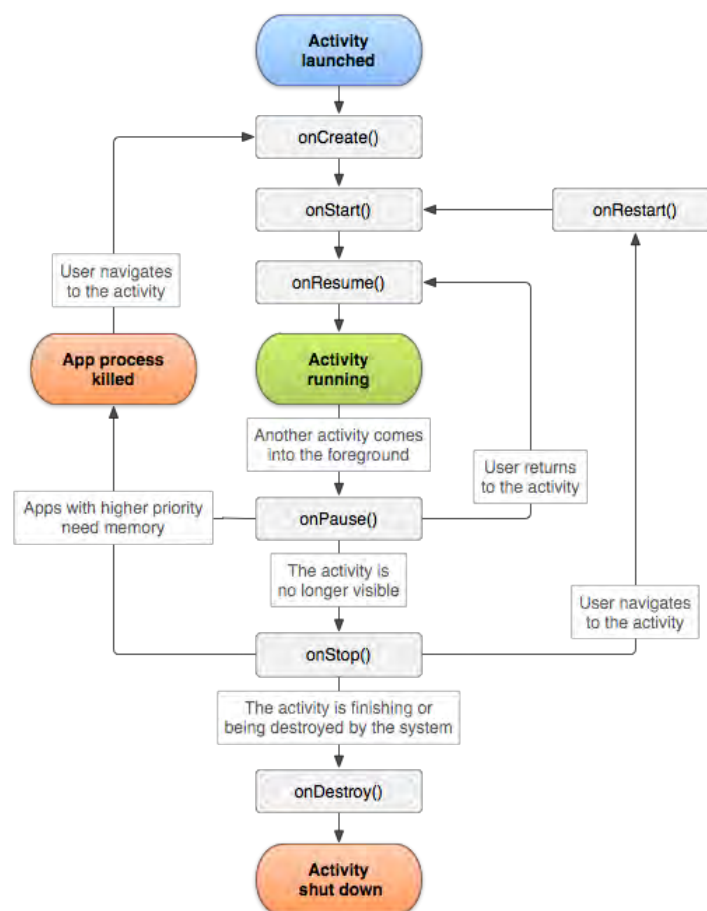
Στην συνέχεια χρησιμοποιώντας το όνομα της μεταβλητής και κάνοντας implement κάθε αναγκαία υπηρεσία το κάθε στοιχείο περνά στον έλεγχο του χρήστη και σχεδιάζεται να υλοποιεί κάποια ενέργεια. Για παράδειγμα παραπάνω για να λειτουργήσει το πάτημα του κουμπιού Button πρέπει να υλοποιηθεί με την συνάρτηση setOnClickListener που θα καταλαβαίνει ποιο κουμπί έχει πατηθεί κάθε φορά και θα ενεργεί αντίστοιχα. Φυσικά πρέπει να γίνει implement και η υπηρεσία OnClickListener στην κλάση για να λειτουργήσει το παραπάνω.

Μια σημαντική λειτουργία που υλοποιείται στις Activities είναι η μετάβαση από την τρέχουσα σε μια άλλη και η προσθήκη διάφορων τιμών μεταβλητών που πρέπει να χρησιμοποιηθούν στην καινούρια. Αυτό γίνεται πολύ εύκολα με τον παρακάτω κώδικα.

```
Intent intent = new Intent(MainActivity.this, ShowTaskEmail.class);
Bundle extras = new Bundle();
extras.putInt("id", curId);
intent.putExtras(extras);
startActivity(intent);
```



Intent είναι η δημιουργία ενός νέου συμβάντος και εδώ συγκεκριμένα θα αρχίσει η Activity με όνομα ShowTaskEmail. Μόλις κλείσει αυτή η Activity και αρχίσει μια άλλη οι τιμές των μεταβλητών χάνονται γι' αυτό είναι σημαντικό να κρατάμε αυτές που θέλουμε και να τις περνάμε στο νέο Intent. Αυτή είναι και η δουλειά του Bundle που παίρνει σαν όρισμα ένα όνομα – κλειδί για να αναγνωρίζεται αργότερα η τιμή της μεταβλητής και την τιμή. Στην συνέχεια το bundle με την putExtras συνάρτηση του Intent περνά την τιμή της μεταβλητής στην νέα Activity όπου μπορεί να την ανακτήσει με την συνάρτηση getIntent().getExtras() του Intent. Επίσης και με το Intent μόνο μπορούμε να περάσουμε μεταβλητές μεταξύ Activities χωρίς την χρήση του Bundle αλλά μόνο με των συναρτήσεων που αναφέραμε πριν του Intent. Παρακάτω δίνεται το διάγραμμα ζωής μιας Activity.

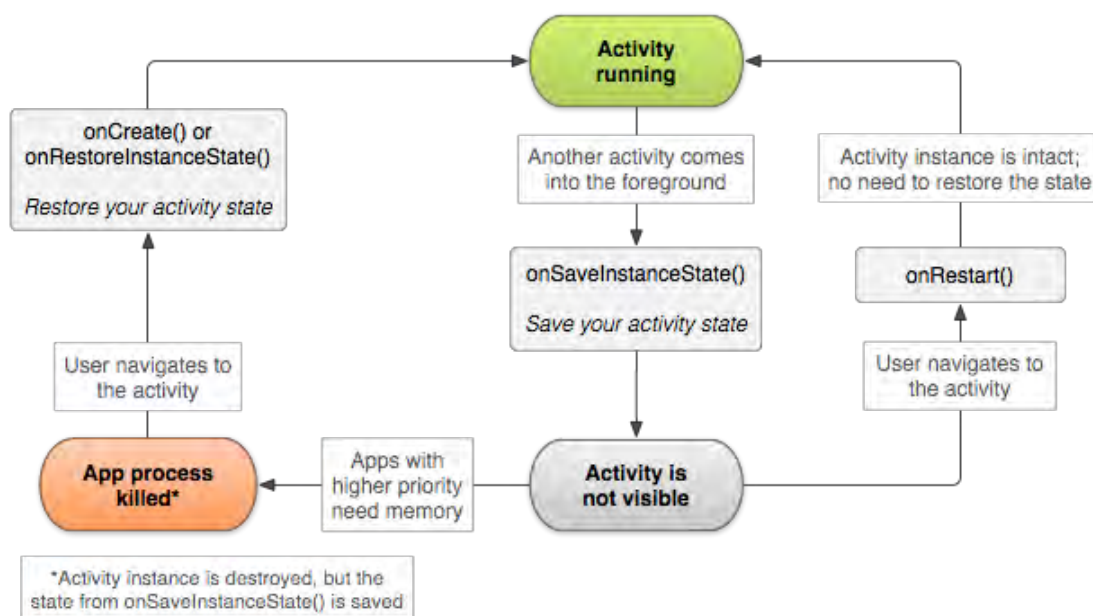


Εικόνα 39. Διάγραμμα ζωής Activity

Όπως φαίνεται στο διάγραμμα όταν η Activity ξεκινά τον κύκλο ζωής της χρησιμοποιεί τις μεθόδους onCreate, onStart και onResume όσο τρέχει. Αν διακοπεί με το onPause τότε δεν καταστρέφεται απλά περιμένει στο background. Αν γίνει onResume, δηλαδή επιστρέψει ο χρήστης πάλι σε αυτή ή την επαναφέρει από το background τότε συνεχίζει κανονικά την λειτουργία της. Σε περίπτωση όμως που κλείσει οριστικά με το onStop τότε αν την ξανανοίξει ο χρήστης θα αρχικοποιηθεί

πάλι ενώ αν την αφήσει ενδεχομένως μετά να έρθει η οριστική καταστροφή της από το σύστημα με την onDestroy.

Σε κάθε περίπτωση υπάρχει τρόπος η Activity να διαφυλάξει την τελευταία της κατάσταση πριν τερματιστεί με το onSaveInstanceState που αποθηκεύει την προηγούμενη της κατάσταση και με την επαναφορά της μετά από έναν τερματισμό γίνεται έλεγχος και αν διαπιστωθεί ότι υπάρχει προηγούμενη αποθηκευμένη κατάσταση κάνει onStart() και επαναφέρεται. Σε περίπτωση που κάνει restart αυτός ο έλεγχος δεν χρειάζεται καθώς η κατάσταση της έχει παραμείνει αναλλοίωτη.



Εικόνα 40. Επαναφορά – Αποθήκευση προηγούμενης κατάστασης Activity

Κάθε activity πρέπει να δηλώνεται μέσα στο manifest αρχείο ανάλογα με τις ιδιότητές της. Αν πρόκειται για την αρχική “MainActivity” που θέλουμε να ανοίγει όταν ξεκινά η εφαρμογή τότε την δηλώνουμε όπως φαίνεται παρακάτω. Όπως παρατηρείτε στο <category> έχει χρησιμοποιηθεί ο χαρακτηρισμός .LAUNCHER υποδηλώνοντάς την ως την αρχική.

```

<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
  
```

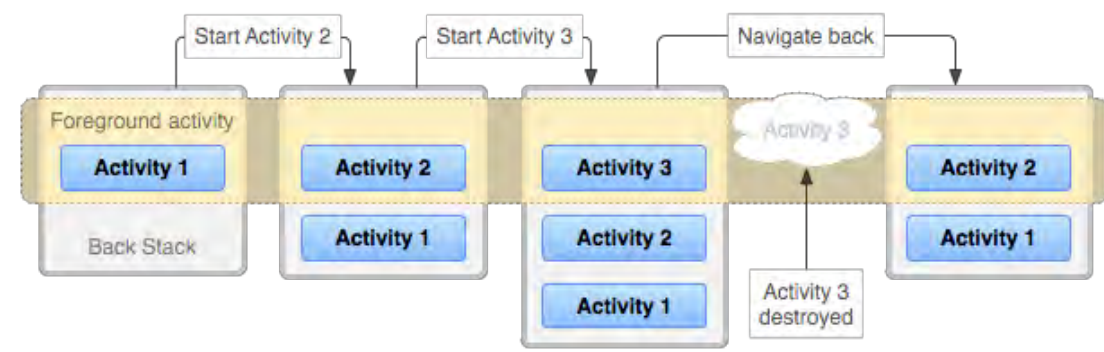
Σε κάθε άλλη περίπτωση η δήλωση είναι διαφορετική καθώς στο <category> χρησιμοποιείται το .DEFAULT όπως φαίνεται στο παράδειγμα παρακάτω. Και στις δύο περιπτώσεις όμως στο <action> χρησιμοποιούμε ένα μοναδικό όνομα που θα ξεχωρίζει την κάθε activity ώστε να χρησιμοποιηθεί στον κώδικα.

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.EXAMPLE" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

Για να ξεκινήσουμε μια activity στον κώδικα Java γράφουμε:

```
Intent intent = new Intent(this, ExampleActivity.class);
startActivity(intent);
```

όπου ExampleActivity το όνομα της activity που θέλουμε να ξεκινήσει. Η Activity που την κάλεσε αν δεν γίνει finish() μπαίνει κάτω από την τρέχουσα στην στοίβα και πατώντας το Back Button επανέρχεται πάλι στο προσκήνιο.



Εικόνα 41. Στοίβα των Activities και navigation με το Back Button

### Αρχεία Services

Ένα Service είναι ένα στοιχείο της εφαρμογής που μπορεί να “τρέχει” στο παρασκήνιο του Android και δεν έχει κάποιο γραφικό περιβάλλον. Πρόκειται για μια κλάση γραμμένη από τον χρήστη που μπορεί να εκτελείται στο background ακόμα και όταν η εφαρμογή έχει τερματιστεί. Ένα service μπορεί να ξεκινήσει μέσα από μια Activity και καθ’ όλη την διάρκεια εκτέλεσής του μπορεί να αλληλεπιδρούν με αυτό και άλλα στοιχεία. Για παράδειγμα ένα service μπορεί να παίζει μουσική, να εκτελεί file I/O, να ελέγχει την τρέχουσα θέση στο GPS κ.α. και όλα αυτά να γίνονται στο παρασκήνιο χωρίς να βλέπει κάτι ο χρήστης.

Όπως και το Activity έτσι και το Service πρέπει να δηλωθεί αναλόγως στο manifest αρχείο. Η δήλωσή του είναι απλή και ένα παράδειγμα φαίνεται παρακάτω.

```
<application>
    ...
    <service android:name=".ExampleService" />
</application>
```

Για να ξεκινήσει το Service να εκτελείται χρησιμοποιείται το startService().

```
Intent intent = new Intent(this, ExampleService.class);
startService(intent);
```

Για να σταματήσει να εκτελείται καλούμε την μέθοδο stopService().

```
package com.uth.remembertodo;
import android.app.IntentService;
public class ScheduledService extends IntentService {
    private NotificationManager mNM;
    public ScheduledService() {
        super("My service");
    }
    @Override
    protected void onHandleIntent(Intent intent) {
        // TODO Auto-generated method stub
    }
}
```

**Εικόνα 42.** Παράδειγμα χρήσης Service από την εφαρμογή για την ειδοποίηση Date and Time

## **Notifications**

Μια Notification είναι ένα μήνυμα που εμφανίζεται στην status bar του κινητού που κατεβαίνει από πάνω προς τα κάτω. Αρχικά εμφανίζεται σαν εικονίδιο στην notification area και στην συνέχεια στο notification drawer στην status bar όπου πέρα από το εικονίδιο περιέχουν και μια σύντομη περιγραφή από δίπλα. Ο χρήστης μπορεί να την διαγράψει απλά τραβώντας την δεξιά ή αριστερά ή να την προβάλει απλά πατώντας πάνω της. Βάση του τελευταίου ξεκινά ένα νέο Intent που ο developer καθορίζει και ξεκινά η αντίστοιχη Activity που θα περιέχει τις πληροφορίες που εμφανίζονταν περιληπτικά στο notification. Στην παρακάτω εικόνα φαίνονται μερικές notifications που έχουν εμφανιστεί στην status bar.



Εικόνα 44. Εμφάνιση ειδοποίησης στη notification area



Εικόνα 43. Ειδοποιήσεις στο notification drawer

Για να δημιουργήσουμε μια notification χρησιμοποιούμε το NotificationManager. Αν ο developer θέλει πατώντας τη notification ο χρήστης να πηγαίνει σε μια καθορισμένη σελίδα στην εφαρμογή τότε πρέπει να δημιουργήσει ένα intent με την καθορισμένη σελίδα και να το περάσει σαν όρισμα στο PendingIntent με flag το .FLAG\_ONE\_SHOT για να χρησιμοποιηθεί μόνο μια φορά. Επίσης πρέπει να ορίσει ένα icon αν το επιθυμεί, τον χρόνο που θα εμφανιστεί και το κείμενο που θα εμφανιστεί στην notification area και στο notification drawer. Ένα παράδειγμα κώδικα από την εφαρμογή που κάνει τα παραπάνω φαίνεται στην παρακάτω εικόνα.

```
mNM = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
Intent notificationIntent = new Intent("com.uth.remembertodo.NOTTASK");
notificationIntent.putExtra("id", id);
PendingIntent pendingIntent = PendingIntent.getActivity(this, id, notificationIntent, PendingIntent.FLAG_ONE_SHOT);

int icon = R.drawable.ic_action_alarms;
String tickerText = "Έχετε μια ειδοποίηση!!";
long when = System.currentTimeMillis();

NotificationCompat.Builder noti = new NotificationCompat.Builder(this);

noti.setTitle(title)
.setContentText(mainBody)
.setTicker(tickerText)
.setSmallIcon(icon)
.setWhen(when);

noti.setDefaults(Notification.DEFAULT_ALL);
noti.setContentIntent(pendingIntent);
noti.setAutoCancel(true);

Notification n = noti.build();

mNM.notify(id, n);
```

Στο notify στην συγκεκριμένη περίπτωση δίνεται το id της κάθε ειδοποίησης που είναι μοναδικό και με αυτό τον τρόπο οι ειδοποιήσεις δεν αντικαθιστούν η μία την άλλη στο notification drawer.

## SQLite Database

Υπάρχουν τέσσερις διαφορετικοί τρόποι για να αποθηκεύει ο developer δεδομένα της εφαρμογής μόνιμα στην εκάστοτε συσκευή:

- **Shared Preferences:** Χρησιμοποιείται κυρίως για την αποθήκευση ρυθμίσεων και μεταβλητών που καταλαμβάνουν λίγο χώρο. Κατά την αποθήκευση με την χρήση των μεθόδων που προσφέρει η SharedPreferences class δίνεται ένα "key" το οποίο χρησιμοποιείται αργότερα κατά την ανάκτηση της τιμής του στοιχείου. Αξίζει να σημειωθεί ότι μπορούν να αποθηκευτούν τιμές διαφόρων τύπων μεταβλητών.
- **Internal Storage:** Αρχεία μπορούν να αποθηκευτούν και στον εσωτερικό χώρο αποθήκευσης της συσκευής. Τα αρχεία αυτά είναι μυστικά και αφορούν την συγκεκριμένη εφαρμογή που τα δημιούργησε. Καμιά άλλη δεν μπορεί να έχει πρόσβαση σε αυτά. Μόλις ο χρήστης απεγκαθηστεί την εφαρμογή αυτά διαγράφονται αυτομάτως. Συνήθως αυτός ο τρόπος αποθήκευσης αφορά μεγάλα αρχεία.
- **External Storage:** Όπως τα αρχεία αποθηκεύονται στον εσωτερικό τόσο μπορούν να αποθηκευτούν και στον εξωτερικό χώρο αποθήκευσης που συνήθως είναι κάποια κάρτα επέκτασης μνήμης. Τα αρχεία αυτά σε αντίθεση με τα άλλα δεν είναι μυστικά και ο χρήστης μπορεί να τα δει αν συνδέσει την συσκευή σε κάποιον υπολογιστή. Αυτός ο τύπος αποθήκευσης δεν ενδείκνυται για όλες τις συσκευές καθώς δεν έχουν όλες υποδοχή για κάρτα μνήμης. Επίσης δεν είναι ιδιαίτερα ασφαλής από την στιγμή που ο καθένας έχει πρόσβαση σε αυτά τα αρχεία. Όπως και προηγουμένως ο τρόπος αυτός αποθήκευσης αφορά κυρίως μεγάλο όγκο αρχείων αδιάφορου ασφαλείας.
- **SQLite Database:** Τέλος ο πιο διαδεδομένος τρόπος αποθήκευσης στοιχείων τύπου πίνακα, με ιδιαίτερη ασφάλεια καθώς αφορά μόνο την εκάστοτε εφαρμογή είναι η χρήση της SQLite Database βάσης δεδομένων. Όπως και στις υπόλοιπες βάσεις δεδομένων έτσι και εδώ η δημιουργία της βασίζεται πάνω στην γλώσσα SQL. Με το SQLiteOpenHelper ο χρήστης δημιουργεί μια database όπως αυτός επιθυμεί με όσους πίνακες και στήλες απαιτείται από την εφαρμογή. Ο τρόπος δημιουργίας είναι standard. Παρακάτω δίνεται ένα ενδεικτικό παράδειγμα δημιουργίας table στην database.

```

public class DictionaryOpenHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 2;
    private static final String DICTIONARY_TABLE_NAME = "dictionary";
    private static final String DICTIONARY_TABLE_CREATE =
        "CREATE TABLE " + DICTIONARY_TABLE_NAME + " (" +
        KEY_WORD + " TEXT, " +
        KEY_DEFINITION + " TEXT);";

    DictionaryOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(DICTIONARY_TABLE_CREATE);
    }
}

```

Τώρα ας δούμε μια εικόνα από την εφαρμογή που χρησιμοποίησα database για την αποθήκευση των διαφόρων ειδοποιήσεων.

```

import java.util.ArrayList;

public class NewDateNotification {

    public static final String KEY_ROWID = "_id";
    public static final String KEY_TITLE = "title";
    public static final String KEY_MAIN = "main";
    public static final String KEY_TIME = "time";
    public static final String KEY_STATUS = "status";
    public static final String KEY_TYPE = "type";
    public static final String KEY_ADDRESS = "address";
    public static final String KEY_LATITUDE = "latitude";
    public static final String KEY_LONGITUDE = "longitude";

    private static final String DATABASE_NAME = "RememberToDoDB";
    private static final String DATABASE_TABLE = "DateNotification";
    private static final int DATABASE_VERSION = 1;

    //To DBHelper αναφέρεται στις 3 παρακάτω classes
    private DBHelper ourHelper;
    private final Context ourContext;
    private SQLiteDatabase ourDatabase;

    private static class DBHelper extends SQLiteOpenHelper {

        public DBHelper(Context context) {
            super(context, DATABASE_NAME, null, DATABASE_VERSION);
            // TODO Auto-generated constructor stub
        }
    }
}

```

Όπως παρατηρούμε έχω ορίσει εννιά διαφορετικές στήλες με τα ονόματά τους, το όνομα της database και το όνομα του ενός table που χρησιμοποίησα. Η database version παίρνει πάντα την τιμή 1 ή 2. Στην συνέχεια δημιουργώ το SQLiteOpenHelper όπως είδαμε και στο παράδειγμα παραπάνω.

```

@Override
public void onCreate(SQLiteDatabase db) {
    // TODO Auto-generated method stub

    //Make database
    db.execSQL("CREATE TABLE " + DATABASE_TABLE + " (" +
        KEY_ROWID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        KEY_TITLE + " TEXT NOT NULL, " +
        KEY_MAIN + " TEXT NOT NULL, " + KEY_TIME + " INTEGER, " +
        KEY_STATUS + " INTEGER, " + KEY_TYPE + " INTEGER, " +
        KEY_ADDRESS + " TEXT NOT NULL, " + KEY_LATITUDE +
        " REAL, " + KEY_LONGITUDE + " REAL);"
    );

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
    onCreate(db);
}
}

```

Με την εντολή CREATE TABLE της SQL δημιουργείται ο πίνακας με το όνομα που έχουμε δηλώσει και τις στήλες ακολουθούμενες από τον τύπο που θα περιέχουν. Το id σε κάθε database είναι Integer Primary Key Autoincrement δηλαδή αυξάνεται μόνο του κάθε φορά και είναι μοναδικό για κάθε στοιχείο μέσα στην βάση. Στην βάση της εφαρμογής οι στήλες πέρα από το id είναι οι:

- **Title:** Περιέχει τον τίτλο της κάθε ειδοποίησης που ελέγχεται πάντα για να είναι μοναδικός.
- **Main:** Περιέχει την περιγραφή της κάθε ειδοποίησης.
- **Time:** Αφορά τις date and time notifications και περιέχει την ώρα ειδοποίησης αλλιώς παίρνει την τιμή 0.
- **Status:** Περιέχει την κατάσταση της κάθε ειδοποίησης σε τιμή μεταβλητής 0 (δεν έχει γίνει ακόμα) και 1 (έχει γίνει).
- **Type:** Περιέχει τον τύπο της κάθε ειδοποίησης σε τιμή μεταβλητής όπου 1 είναι date and time notification και 2 gps notification.
- **Address:** Αφορά τις gps notifications και αποθηκεύει την διεύθυνση ειδοποίησης αλλιώς παίρνει την τιμή null.
- **Latitude:** Αφορά τις gps notifications και αποθηκεύει το γεωγραφικό μήκος από τους χάρτες αλλιώς χρησιμοποιείται στις date and time notifications σαν



μεταβλητή κατάστασης συγχρονισμού, δηλαδή 0 (δεν έχει συγχρονιστεί με το Ημερολόγιο αυτή η ειδοποίηση) και 1 (έχει συγχρονιστεί).

- **Longitude:** Αφορά τις gps notifications και αποθηκεύει το γεωγραφικό πλάτος από τους χάρτες αλλιώς παίρνει την τιμή 0.

Μέσα στην μέθοδο onUpgrade() εκτελείται η εντολή DROP TABLE IF EXISTS που διαγράφει τον πίνακα από την database όταν εκτελεστεί. Ένα παράδειγμα για το πώς λειτουργεί μια μέθοδο στην database δίνεται στην εικόνα παρακάτω που ψάχνει να δει αν υπάρχουν GPS ειδοποιήσεις αποθηκευμένες και επιστρέφει boolean τιμή.

```
public boolean hasGPSNotis() {
    // TODO Auto-generated method stub
    String[] columns = new String[] { KEY_ROWID, KEY_TITLE, KEY_MAIN, KEY_TIME, KEY_STATUS, KEY_TYPE, KEY_ADDRESS, KEY_LATITUDE, KEY_LONGITUDE };
    Cursor c = ourDatabase.query(DATABASE_TABLE, columns, KEY_STATUS + "=" + 0 + " AND " + KEY_TYPE + "=" + 2, null, null, null, null);
    if (c.moveToFirst()) {
        return true;
    }
    return false;
}
```

Παρατηρούμε ότι με το query διατρέχουμε τον πίνακα της database και με τις προϋποθέσεις που παίρνει σαν τρίτο όρισμα επιστρέφει τα ανάλογα αποτελέσματα. Λειτουργεί όπως το SELECT της SQL. Τα αποτελέσματα αποθηκεύονται σε μια μεταβλητή τύπου Cursor και μετά ελέγχω αν κάνει moveToFirst, δηλαδή αν έχει επιστρέψει αποτελέσματα το query, τότε υπάρχουν GPS notifications αλλιώς όχι.

Τέλος υπάρχει τρόπος και απευθείας χρήσης της γλώσσας SQL και όχι μέσα από κάποια μέθοδο της SQLite. Παρακάτω φαίνεται ο τρόπος αυτός όπου με το SELECT επιλέγει όλες τις ειδοποιήσεις – στοιχεία της βάσης ταξινομημένα με αύξουσα σειρά ανάλογα με τον τύπο ειδοποίησης Type. Αυτή η μέθοδος επιστρέφει μια λίστα από αντικείμενα – ειδοποιήσεις που εμφανίζονται στην αρχική οθόνη της εφαρμογής.

```
public List<Task> getAllTasks() {
    List<Task> taskList = new ArrayList<Task>();
    // Select All Query
    String selectQuery = "SELECT * FROM " + DATABASE_TABLE + " ORDER BY " + KEY_STATUS + " ASC";
    Cursor cursor = ourDatabase.rawQuery(selectQuery, null);
    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {
            Task task = new Task();
            task.setId(cursor.getInt(0));
            task.setTaskTitle(cursor.getString(1));
            task.setTaskMain(cursor.getString(2));
            //task.setTaskTime(cursor.getLong(3));
            task.setStatus(cursor.getInt(4));
            task.setType(cursor.getInt(5));
            // Adding contact to list
            taskList.add(task);
        } while (cursor.moveToNext());
    }
    // return task list
    return taskList;
}
```

## Widgets

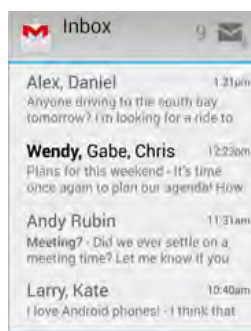
Τα widgets είναι μια “μικρή εικόνα” της εκάστοτε εφαρμογής που περιέχει κάποια γεγονότα από αυτή ή χρησιμοποιούνται για πιο σύντομη και γρήγορη χρήση της. Εμφανίζονται στην home screen της συσκευής και μπορούν να μεταφερθούν από τον χρήστη σε οποιοδήποτε σημείο αυξομειώνοντας ακόμα και το μέγεθός τους.

Υπάρχουν τρία βασικά ήδη widget:

- **Information Widgets:** Τα information widgets είναι ο πιο απλός τύπος. Εμφανίζουν πληροφορίες που ενδιαφέρουν τον χρήστη και ανανεώνονται σε τακτά χρονικά διαστήματα. Τέτοιου τύπου είναι τα widget καιρού, ρολόγια, αθλητικής ενημέρωσης κτλ. Πατώντας τα ανοίγει η αντίστοιχη εφαρμογή και εμφανίζονται λεπτομερώς οι πληροφορίες που φαίνονται σε αυτά.
- **Collection Widgets:** Σε αυτό τον τύπο widget εμφανίζονται, όπως μαρτυρά και το όνομά τους, μια ομάδα αντικειμένων του ίδιου τύπου όπως εικόνες, email ή και μια συλλογή από άρθρα από μια εφαρμογή ενημέρωσης. Σε αυτά ο χρήστης μπορεί να κάνει scroll κάθετα για να τα δει όλα μέσα από μια σύντομη περιήγηση ανάμεσα στα στοιχεία που εμφανίζονται και επιλέγοντας κάποιο να εμφανίσει λεπτομερώς τα στοιχεία για αυτό.
- **Control Widgets:** Βασικός σκοπός αυτού του τύπου είναι η απεικόνιση λειτουργιών που χρησιμοποιεί συχνότερα ο χρήστης και μπορούν να ενεργοποιηθούν άμεσα από την home screen χωρίς να χρειάζεται να ανοίξει η αντίστοιχη εφαρμογή. Λειτουργούν σαν “τηλεχειριστήριο” για την εφαρμογή. Η χρήση τους δεν απαιτεί την εκκίνηση της εφαρμογής ή την εμφάνιση κάποιας activity με λεπτομερή στοιχεία όπως γινόταν στους δύο προηγούμενους τύπους. Τέτοιου τύπου widget είναι π.χ. ένα μουσικό widget με τα κουμπιά play-pause-next ή και αυτό που χρησιμοποιείται πιο συχνά που σου δίνει την δυνατότητα να ενεργοποιήσεις κάποιες ρυθμίσεις απευθείας χωρίς να χρειάζεται να ανοίξεις την εφαρμογή του Android.



Εικόνα 47. Information Widget



Εικόνα 46. Collection Widget



Εικόνα 45. Control Widget

Για την χρήση του widget στην εφαρμογή απαραίτητη είναι η δήλωσή του στο Android Manifest όπως φαίνεται παρακάτω

```
<receiver android:name="ExampleAppWidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/example_appwidget_info" />
</receiver>
```

όπου το AppWidgetManager είναι υπεύθυνο για την λειτουργία του widget και στέλνει όλα τα σήματα από αυτό στο AppWidgetProvider.

Το AppWidgetProviderInfo καθορίζει τις μεταβλητές για το widget όπως το ελάχιστο πλάτος και ύψος, πόσο συχνά να κάνει update και προαιρετικά την εκκίνηση μιας Activity μόλις ο χρήστης αλληλεπιδράσει με αυτό. Αυτές οι μεταβλητές καταχωρούνται σε ένα xml αρχείο τύπου <appwidget-provider> στην τοποθεσία res/xml/ στο project και έχουν την παρακάτω ενδεικτική μορφή

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="40dp"
    android:minHeight="40dp"
    android:updatePeriodMillis="86400000"
    android:previewImage="@drawable/preview"
    android:initialLayout="@layout/example_appwidget"
    android:configure="com.example.android.ExampleAppWidgetConfigure"
    android:resizeMode="horizontal|vertical"
    android:widgetCategory="home_screen|keyguard"
    android:initialKeyguardLayout="@layout/example_keyguard">
</appwidget-provider>
```

Όπως προαναφέραμε μπορούμε να καθορίσουμε πολλές παραμέτρους του widget στο παραπάνω αρχείο. Μερικές βασικές είναι το previewImage όπου δίνουμε στον χρήστη μια εικόνα του widget πριν το βάλει στην homescreen, το InitialLayout που είναι το xml του widget μας δηλαδή το γραφικό του περιβάλλον, το configure όπου δηλώνουμε το Activity που σχετίζεται με το widget και κάποιες έξτρα παράμετροι που δεν είναι πάντα απαραίτητοι.

Τέλος για να λειτουργήσει το widget δημιουργούμε την AppWidgetProvider Class όπου μέσα από κάποιες μεθόδους όπως την onUpdate(), onReceive(), onAppWidgetOptionsChanged() καθορίζουμε την λειτουργία του. Βασική μέθοδος είναι η onUpdate() που όπως δηλώνει και το όνομά της είναι υπεύθυνη για την ανανέωση του περιεχομένου του με βάση τον χρόνο που έχουμε δηλώσει στην AppWidgetProviderInfo παραπάνω ή χειροκίνητα πατώντας κάποιο κουμπί. Παρακάτω δίνεται κάποιο ενδεικτικό παράδειγμα κώδικα

```

public class ExampleAppWidgetProvider extends AppWidgetProvider {

    public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds)
        final int N = appWidgetIds.length;

    // Perform this loop procedure for each App Widget that belongs to this provider
    for (int i=0; i<N; i++) {
        int appWidgetId = appWidgetIds[i];

        // Create an Intent to launch ExampleActivity
        Intent intent = new Intent(context, ExampleActivity.class);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);

        // Get the layout for the App Widget and attach an on-click listener
        // to the button
        RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.appwidget_pro
        views.setOnClickPendingIntent(R.id.button, pendingIntent);

        // Tell the AppWidgetManager to perform an update on the current app widget
        appWidgetManager.updateAppWidget(appWidgetId, views);
    }
}
}
}

```

Για την εκκίνηση κάποιας Activity δημιουργείται ένα PendingIntent και δεν γίνεται όπως συνήθως. Η δήλωση της Activity έχει την παρακάτω μορφή στο manifest αρχείο

```

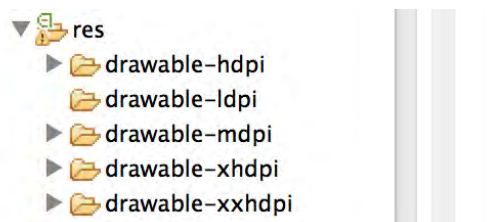
<activity android:name=".ExampleAppWidgetConfigure">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_CONFIGURE"/>
    </intent-filter>
</activity>

```

### Εικόνες και άλλα αρχεία της εφαρμογής

Κάθε εφαρμογή έχει ένα χαρακτηριστικό εικονίδιο και διάφορες άλλες εικόνες. Για να φαίνονται όμως σε κάθε συσκευή που τρέχει Android φυσιολογικά, καθώς οι αναλύσεις διαφέρουν, πρέπει να εισάγει ο developer την εικόνα με το ίδιο όνομα αλλά με διαφορετικές αναλύσεις στους κατάλληλους φακέλους. Οι φάκελοι αυτοί βρίσκονται στον φάκελο res/ του project και ονομάζονται drawable. Ανάλογα με τις αναλύσεις τους οι εικόνες πρέπει να βρίσκονται στον σωστό φάκελο καθώς η εφαρμογή κάθε φορά καταλαβαίνει την ανάλυση της κάθε συσκευής και χρησιμοποιεί την αντίστοιχη ανάλυση.

Παρακάτω φαίνονται οι φάκελοι όπως βρίσκονται μέσα στο project



όπου οι χαμηλότερες αναλύσεις 320x480 μπαίνουν στον φάκελο ldpi (low density pixel) και οι υπόλοιπες με την σειρά στον mdpi, hdpi, xhdpi και xxhdpi που μπαίνουν οι εικόνες μεγάλης ανάλυσης τύπου 1920x1080 FullHD.

Οποιοδήποτε άλλο αρχείο μπαίνει σε φάκελο μέσα στο res/ όπως για παράδειγμα τα αρχεία ήχου μέσα στον φάκελο raw σε μορφή mp3. Για να τα χρησιμοποιήσει αυτά ο developer στον κώδικα πρέπει να κάνει αναφορά στον φάκελο που βρίσκονται. Στις εικόνες για παράδειγμα χρησιμοποιείται το @drawable/image στον xml κώδικα και το R.drawable.image σε Java κώδικα.

### **Αρχείο AndroidManifest**

Ένα από τα πιο βασικά αρχεία του project είναι το Manifest αρχείο που παρέχει βασικές πληροφορίες για την εφαρμογή στο σύστημα Android καθώς και απαιτήσεις που πρέπει να έχει το σύστημα πριν εκτελέσει την εφαρμογή.

Αναλυτικότερα:

- Καθορίζει το όνομα του πακέτου Java που είναι μοναδικό για κάθε project και αποτελεί το ID του.
- Περιγράφει τα περιεχόμενα της εφαρμογής όπως τα Activities, τα permissions, τα services, τα broadcast receivers και τα content providers από τα οποία αποτελείται. Επίσης περιέχει τα ονόματα των κλάσεων που περιέχει κάθε component και τον τύπο τους. Με βάση αυτών των στοιχείων το Android σύστημα είναι σε θέση να γνωρίζει ποια είναι τα components και που χρησιμοποιούνται.
- Καθορίζει ποιες διεργασίες θα ασχοληθούν με τα components της εφαρμογής.
- Περιλαμβάνει τα permissions που χρειάζεται η εφαρμογή για να χρησιμοποιήσει διάφορα parts του API ή για να επικοινωνήσει με άλλα applications.
- Επίσης καθορίζει την κατώτατη έκδοση Android για την οποία θα λειτουργεί η εφαρμογή.
- Τέλος απαριθμεί τις βιβλιοθήκες με τις οποίες η εφαρμογή πρέπει να είναι συνδεδεμένη.

Η ενδεικτική δομή του Manifest αρχείου δίνεται παρακάτω και περιλαμβάνει όσα προαναφέραμε και χρησιμοποιήθηκαν στην εφαρμογή.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.uth.rembertodo"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true"/>

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />

    <uses-permission android:name="com.uth.rembertodo.permission.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
    <uses-permission android:name="android.permission.READ_CALENDAR" />
    <uses-permission android:name="android.permission.WRITE_CALENDAR" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/rem"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

```

Με μια σύντομη ματιά και όσα αναφέραμε παραπάνω γίνεται φανερό ότι η εφαρμογή έχει κατώτατο όριο την έκδοση 14 του συστήματος Android, περιλαμβάνει permissions για πολλές υπηρεσίες που παρέχει το API όπως Internet, Google Maps, GPS, Calendar (Ημερολόγιο), Boot Services (Έλεγχος Επανεκκίνησης Συσκευής) και Storage (Εγγραφή σε εξωτερική μνήμη). Ακόμη δηλώνεται η χρήση της OpenGL, δηλώνονται τα γραφικά στοιχεία της εφαρμογής όπως εικονίδια κτλ και τέλος η ενεργοποίηση των υπηρεσιών Google Play Services που χρειάζονται για τους χάρτες.

## 4. Επίδειξη Εφαρμογής RememberToDo

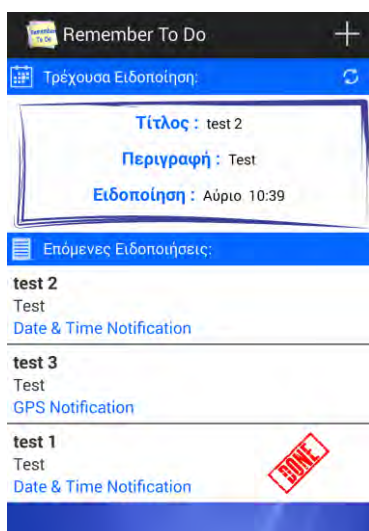
### 4.1 Βασικές λειτουργίες της εφαρμογής



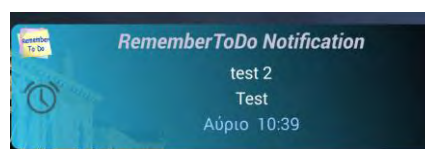
Όπως έχουμε αναφέρει και στην εισαγωγή βασική λειτουργία της εφαρμογής είναι η υπενθύμιση του χρήστη για κάποιο γεγονός που επιθυμεί με δύο διαφορετικούς τρόπους:

- Με βάση την ημερομηνία και ώρα
- Με βάση την γεωγραφική του θέση

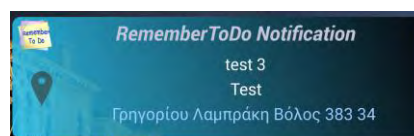
Αρχικά κατά την εκκίνηση της εφαρμογής εμφανίζεται στον χρήστη η παρακάτω οθόνη. Στο λευκό πλαίσιο εμφανίζεται, αν υπάρχει, η τρέχουσα ειδοποίηση, αυτή δηλαδή που θα γίνει πιο σύντομα. Αν υπάρχουν ενεργές ειδοποιήσεις ημερομηνίας αλλά και με βάση το GPS τότε στο πλαίσιο θα φαίνεται πάντα η πιο σύντομη από τις τρέχουσες ειδοποιήσεις ημερομηνίας. Σε περίπτωση όμως που υπάρχουν μόνο ειδοποιήσεις τοποθεσίας ενεργές τότε η πιο παλιά από αυτές εμφανίζονται στο πλαίσιο. Το ίδιο ισχύει και για το widget της εφαρμογής.




Εικόνα 50. Αρχική οθόνη RememberToDo





Εικόνα 48. Widget με ειδοποίηση ημερομηνίας

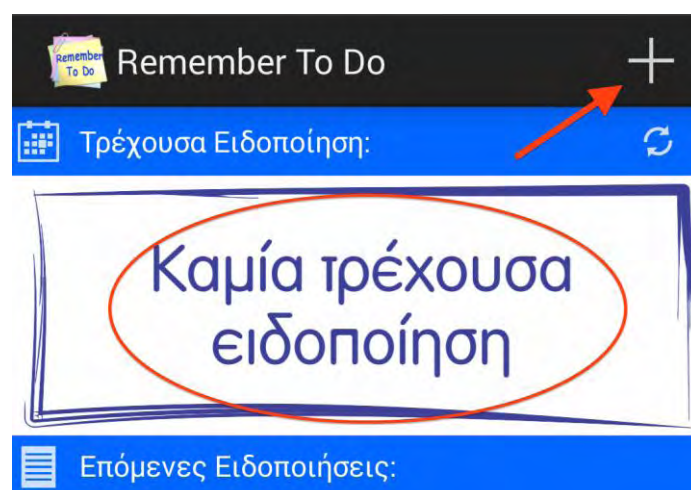


Εικόνα 49. Widget με ειδοποίηση τοποθεσίας

Κάτω ακριβώς από το πλαίσιο της τρέχουσας ειδοποίησης βρίσκεται το πλαίσιο της λίστας με τις επόμενες ειδοποιήσεις. Εκεί βρίσκονται όλες οι ειδοποιήσεις που πρόκειται να γίνουν ή που έχουν γίνει και απλά είναι αποθηκευμένες με την ένδειξη DONE. Για κάθε ειδοποίηση στην λίστα φαίνεται ο τίτλος, η περιγραφή και ο τύπος της. Πατώντας πάνω σε κάποια από αυτές ανοίγει μια νέα καρτέλα με τα στοιχεία της ειδοποίησης και δίνεται στον χρήστη η δυνατότητα να την επεξεργαστεί ή να την διαγράψει. Επίσης πατώντας παρατεταμένα πάνω σε οποιαδήποτε από αυτές εμφανίζεται στην άκρη ένα κόκκινο μικρό εικονίδιο κάδου.  Πατώντας το διαγράφεται η αντίστοιχη ειδοποίηση.

Σε περίπτωση κάποιας αλλαγής ή κολλήματος της εφαρμογής ο χρήστης μπορεί να πατήσει το κουμπί της ανανέωσης (refresh)  που βρίσκεται πάνω δεξιά και έτσι να γίνει ανανέωση της λίστας και πιθανόν να διορθωθεί το πρόβλημα στις ειδοποιήσεις.

Η δημιουργία νέας ειδοποίησης γίνεται στην καρτέλα που εμφανίζεται μόλις ο χρήστης πατήσει το κουμπί  στην αρχική οθόνη της εφαρμογής για να δημιουργήσει μια νέα ειδοποίηση. Επίσης αντί γι' αυτό το κουμπί ο χρήστης μπορεί να πατήσει το πλαίσιο που αναφέρει "Καμιά Τρέχουσα Ειδοποίηση", σε περίπτωση που δεν υπάρχει κάποια ενεργή, για να του ανοίξει η καρτέλα επιλογής τύπου. Το ίδιο μπορεί να γίνει και από το widget της εφαρμογής, για την ίδια περίπτωση, πατώντας μια φορά πάνω του.

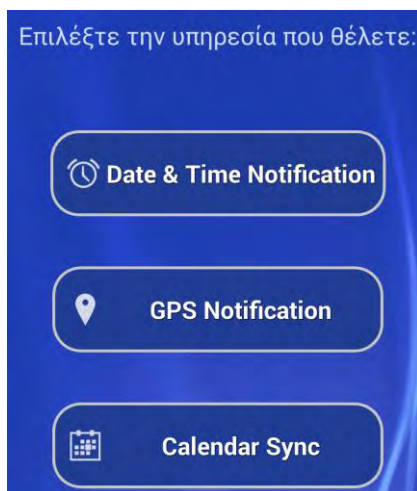


Εικόνα 51. Δημιουργία νέας ειδοποίησης

Στην συνέχεια εμφανίζεται στον χρήστη η παρακάτω καρτέλα που πρέπει να επιλέξει τον τύπο που επιθυμεί, δηλαδή Date and Time (Ημερομηνίας - Ώρας) ή GPS (Γεωγραφικής Θέσης - Τοποθεσίας). Ακόμη επιλέγοντας την τρίτη επιλογή Calendar Sync (Συγχρονισμός Ειδοποιήσεων Ημερολογίου) μπορεί να επιλέξει τις Date and



Time ειδοποιήσεις που θέλει να συγχρονίσει με την stock εφαρμογή ημερολογίου του Android.



Εικόνα 52. Επιλογή τύπου ειδοποίησης

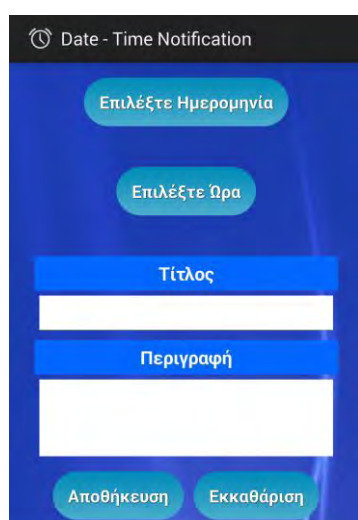
#### 4.1.1 Ειδοποίηση Ημερομηνίας (Date & Time Notification)

Σε αυτό τον τύπο ειδοποίησης ο χρήστης επιλέγει την ημερομηνία και την ώρα που θέλει να ειδοποιηθεί και γράφει τον τίτλο και την περιγραφή της ειδοποίησης του. Όλα τα πεδία είναι απαραίτητα για την δημιουργία της καθώς και η επιλογή της ώρας πρέπει να είναι κατά τρία λεπτά μεγαλύτερη από την τρέχουσα για να είναι έγκυρη.

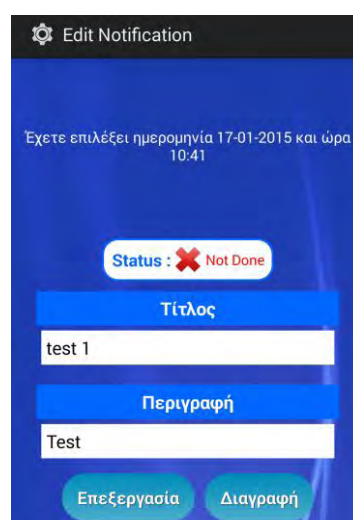
Κατά την επιλογή της ημερομηνίας και της ώρας υπάρχει ορθολογικός έλεγχος πέρα από την διαφορά των τριών λεπτών. Αν η ημερομηνία που επιθυμεί να επιλέξει ο χρήστης έχει περάσει τότε εμφανίζεται μήνυμα λάθους. Το ίδιο συμβαίνει και με την επιλογή της ώρας. Σε περίπτωση που ο χρήστης επιλέξει έγκυρα ημερομηνία και ώρα αλλά δεν αποθηκεύσει την ειδοποίηση άμεσα και η ορισμένη ώρα έχει περάσει πάλι θα εμφανιστεί αντίστοιχο μήνυμα λάθους καθώς γίνεται και πάλι έλεγχος πατώντας το κουμπί της αποθήκευσης. Τέλος ελέγχεται και ο τίτλος της αν υπάρχει για να μην ξαναχρησιμοποιηθεί.

Αν όλα οριστούν σωστά τότε η ειδοποίηση αποθηκεύεται στην βάση δεδομένων της εφαρμογής με ένα μοναδικό id το οποίο και χρησιμοποιείται για την εύρεση και την επεξεργασία της σε επίπεδο κώδικα, τον μοναδικό της τίτλο, την περιγραφή, την ώρα ειδοποίησης που προκύπτει από την ημερομηνία και την ώρα που έχει επιλέξει σε μορφή milliseconds και τέλος μια μεταβλητή που φανερώνει αν η ειδοποίηση είναι ή όχι ενεργή.

Εφόσον έχει αποθηκευτεί μπορεί να υποστεί επεξεργασία σε επιτρεπτά πλαίσια ανά πάσα στιγμή από την αρχική οθόνη της εφαρμογής πατώντας πάνω της και Επεξεργασία. Αμέσως εμφανίζονται στην καρτέλα τα κουμπιά για την επιλογή ημερομηνίας - ώρας και ο τίτλος - περιγραφή ανοίγουν ξανά προς επεξεργασία αν το επιθυμεί ο χρήστης. Όπως και στην δημιουργία νέας, έτσι και εδώ υπάρχουν οι ίδιοι έλεγχοι. Μετά τις αλλαγές πατώντας το κουμπί Αποθήκευση, ανανεώνονται τα στοιχεία της ειδοποίησης στην βάση. Σε περίπτωση που η ειδοποίηση έχει γίνει και είναι ανενεργή, πατώντας Επεξεργασία και δίνοντας μια έγκυρη ημερομηνία και ώρα ενεργοποιείται ξανά. Τέλος πατώντας το κουμπί Διαγραφή η ειδοποίηση διαγράφεται και αφαιρείται από την βάση δεδομένων.




Εικόνα 53. Δημιουργία νέας ειδοποίησης Ημερομηνίας κ' Ώρας

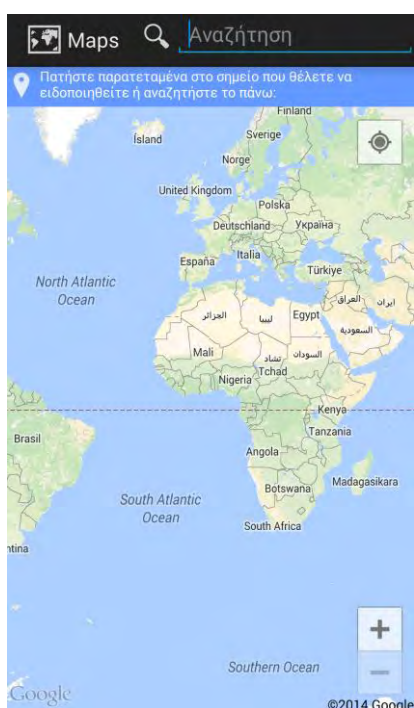


Εικόνα 54. Περιγραφή – Επεξεργασία – Διαγραφή ειδοποίησης Ημερομηνίας κ' Ώρας

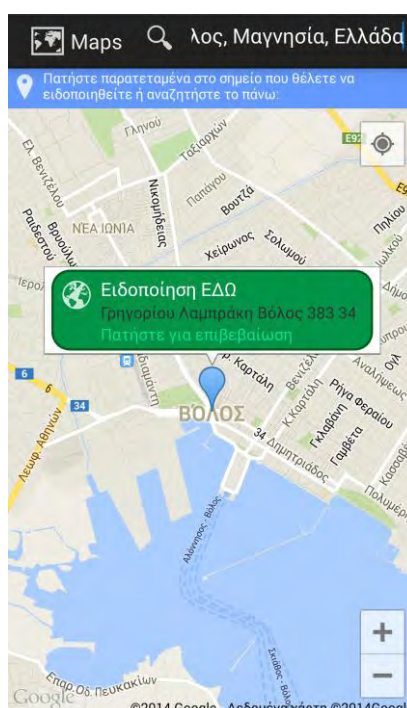
#### 4.1.2 Ειδοποίηση Τοποθεσίας (GPS Notification)

Ο δεύτερος τύπος ειδοποίησης αφορά αυτή με βάση την τοποθεσία στον χάρτη. Μόλις επιλεγεί αυτός ο τύπος εμφανίζεται στον χρήστη ο παγκόσμιος χάρτης της Google. Βασική προϋπόθεση λειτουργίας των χαρτών είναι η ύπαρξη σύνδεσης στο Internet. Σε περίπτωση που δεν υπάρχει ζητείται από τον χρήστη να μεταβεί στις ρυθμίσεις να το ενεργοποιήσει. Όταν ανοίξει ο χάρτης, πατώντας το κουμπί τρέχουσας θέσης πάνω δεξιά  γίνεται zoom στην θέση που βρίσκεται εκείνη την στιγμή ο χρήστης. Όπως και για την σύνδεση στο Internet έτσι και εδώ, για λειτουργήσει η επιλογή θέσης στον χάρτη ή η αναζήτηση τοποθεσίας από πάνω στην ActionBar είναι απαραίτητη η ενεργοποίηση των υπηρεσιών τοποθεσίας της συσκευής. Σε περίπτωση που δεν είναι ενεργοποιημένες εμφανίζεται αντίστοιχο μήνυμα και ζητείται από τον χρήστη να μεταβεί στις ρυθμίσεις να τις ενεργοποιήσει.

Υπάρχουν δύο διαφορετικοί τρόποι για να επιλεγεί η επιθυμητή τοποθεσία ειδοποίησης στον χάρτη. Αρχικά πατώντας παρατεταμένα πάνω του, στο σημείο το οποίο επιθυμεί να ειδοποιηθεί ο χρήστης, ή αναζητώντας την πόλη – οδό στο κουτάκι πάνω από τον χάρτη. Και στις δύο περιπτώσεις εμφανίζεται το γαλάζιο marker με το InfoWindow από πάνω με μια σύντομη περιγραφή του σημείου. Ο χρήστης στην συνέχεια πρέπει να πατήσει πάνω σε αυτό για να μεταβεί στην επόμενη καρτέλα που είναι η συμπλήρωση του τίτλου και της περιγραφής της ειδοποίησης. Το marker μπορεί να μετακινηθεί πάνω στον χάρτη και να τοποθετηθεί σε κάποια άλλη θέση αν χρειαστεί.



**Εικόνα 56.** Χάρτες Google για ειδοποίηση τοποθεσίας

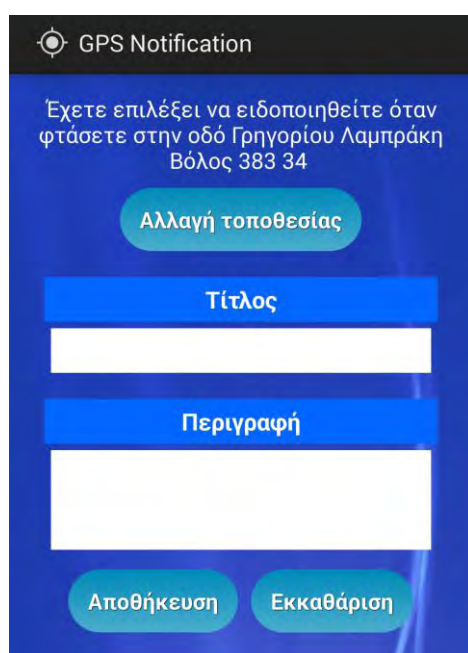


**Εικόνα 55.** Ειδοποίηση τοποθεσίας με αναζήτηση

Όπως και στις ειδοποιήσεις ημερομηνίας έτσι και εδώ υπάρχουν διάφοροι έλεγχοι ορθής λειτουργίας. Όπως προαναφέραμε ελέγχεται αρχικά η σύνδεση στο Internet που είναι απαραίτητη για την λειτουργία των χαρτών και η χρήση του GPS της συσκευής που απαιτείται για την εύρεση της τρέχουσας θέσης του χρήστη. Αν κατά την χρήση των χαρτών διακοπή για κάποιο λόγο η σύνδεση στο διαδίκτυο τότε εμφανίζεται αντίστοιχο μήνυμα σφάλματος και τερματίζεται η λειτουργία τους. Σε επίπεδο ορθής επιλογής τοποθεσίας δεν επιτρέπεται στον χρήστη να επιλέξει τοποθεσία για ειδοποίηση που απέχει λιγότερο από 10 μέτρα από την τρέχουσα θέση του.

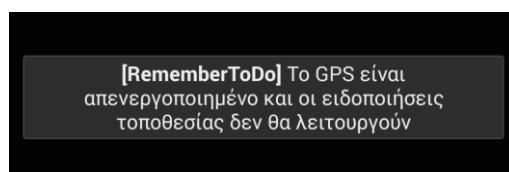
Αφού επιλεγεί η τοποθεσία και πατήσει ο χρήστης πάνω στο InfoWindow ανοίγει η επόμενη καρτέλα όπου δίνεται ο τίτλος και η περιγραφή της ειδοποίησης. Ακόμη υπάρχει και ένα κουμπί αλλαγής τοποθεσίας ειδοποίησης όπου αν πατηθεί γυρνά τον χρήστη στους χάρτες και στην επιλεγμένη του θέση ώστε να την αλλάξει

ή να επιλέξει μια άλλη. Τέλος μόλις αποθηκευτεί η ειδοποίηση στην βάση δεδομένων ξεκινά η συνεχής παρακολούθηση της τρέχουσας θέσης από το GPS μέχρι να φτάσει ο χρήστης στο σημείο ειδοποίησης. Κάθε ειδοποίηση τοποθεσίας έχει το δικό της id, τον μοναδικό της τίτλο όπως και οι ειδοποιήσεις ημερομηνίας, την περιγραφή της, το γεωγραφικό μήκος και γεωγραφικό πλάτος την τοποθεσίας ειδοποίησης, το όνομα της οδού και μια μεταβλητή που δηλώνει αν είναι ή όχι ενεργή.



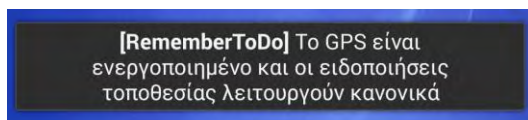
Εικόνα 57. Δημιουργία ειδοποίησης τοποθεσίας

Από την στιγμή που αποθηκευτεί έστω και μια ειδοποίηση τοποθεσίας ελέγχεται συνεχώς από το GPS η τρέχουσα τοποθεσία του χρήστη. Σε περίπτωση τερματισμού της εφαρμογής διακόπτεται η χρήση του GPS και η ειδοποίηση δεν θα γίνει. Αυτό γίνεται για λόγους οικονομίας μπαταρίας της συσκευής καθώς η χρήση των υπηρεσιών τοποθεσίας την εξασθενούν γρήγορα. Όταν όμως η εφαρμογή τρέχει στο παρασκήνιο το GPS ελέγχει τακτικά. Η απενεργοποίηση των υπηρεσιών τοποθεσίας σε αυτή την περίπτωση οδηγεί στην εμφάνιση ενός προειδοποιητικού μηνύματος από την εφαρμογή όπως φαίνεται παρακάτω.



Εικόνα 58. Προειδοποιητικό μήνυμα διακοπής του GPS όταν υπάρχει ειδοποίηση τοποθεσίας

Αν ο χρήστης ενεργοποιήσει ξανά τις υπηρεσίες τοποθεσίας τότε το GPS συνεχίζει να ελέγχει την τρέχουσα θέση και εμφανίζεται πάλι μήνυμα ότι όλα λειτουργούν ξανά.

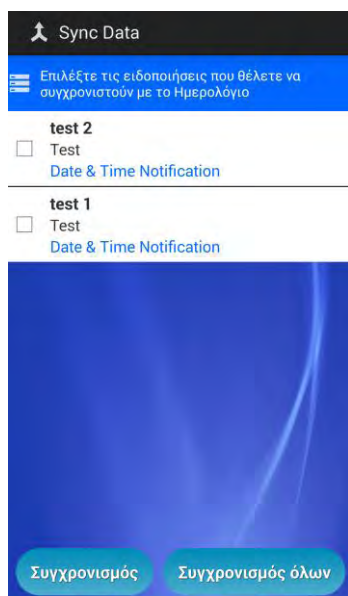


Εικόνα 59. Προειδοποιητικό μήνυμα ότι το GPS λειτουργεί ξανά

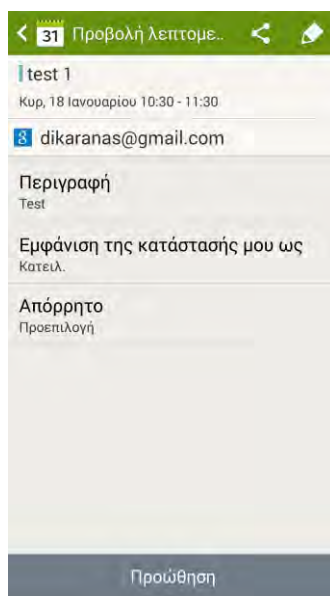
### 4.1.3 Συγχρονισμός ειδοποιήσεων με Ημερολόγιο (Calendar Sync)

Τρίτη και τελευταία λειτουργία της εφαρμογής είναι ο συγχρονισμός των ειδοποιήσεων ημερομηνίας με την stock εφαρμογή Ημερολογίου του Android. Στην καρτέλα αυτή ο χρήστης επιλέγει ποιες ειδοποιήσεις θέλει να συγχρονίσει με το Ημερολόγιο και πατώντας το κουμπί αυτές καταχωρούνται αυτόματα σε αυτό. Σε περίπτωση που ο χρήστης διαθέτει Google Calendar και το έχει συνδέσει με το Ημερολόγιο οι ειδοποιήσεις καταχωρούνται και σε αυτό και ειδοποιείται με email την ώρα ειδοποίησης.

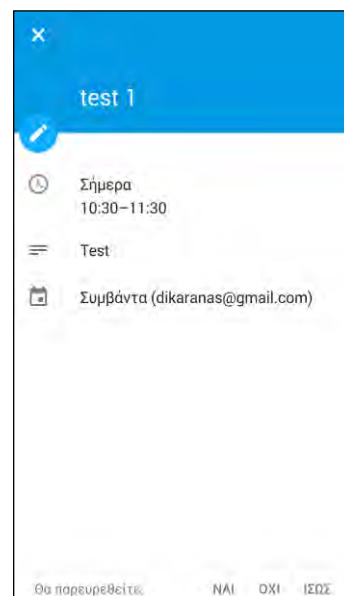
Οι ειδοποιήσεις που δεν έχουν γίνει ακόμα καταχωρούνται με περίοδο ειδοποίησης μιας ώρας πριν από την ώρα που έχει οριστεί από τον χρήστη. Οπότε υπάρχει περιθώριο μιας ώρας να ειδοποιηθεί από την σχετική εφαρμογή πέρα από την RememberToDo. Οι ειδοποιήσεις που έχουν γίνει και καταχωρούνται με την σειρά τους στο Ημερολόγιο, διαθέτουν ώρα ειδοποίησης την ώρα που είχε οριστεί και έγινε η ειδοποίηση. Οι ειδοποιήσεις που έχουν συγχρονιστεί δεν ξαναεμφανίζονται στην λίστα επιλογής συγχρονισμού. Παρακάτω φαίνεται μια ενδεικτική λίστα στην καρτέλα συγχρονισμού.



Εικόνα 62. Λίστα συγχρονισμού ειδοποιήσεων ημερομηνίας



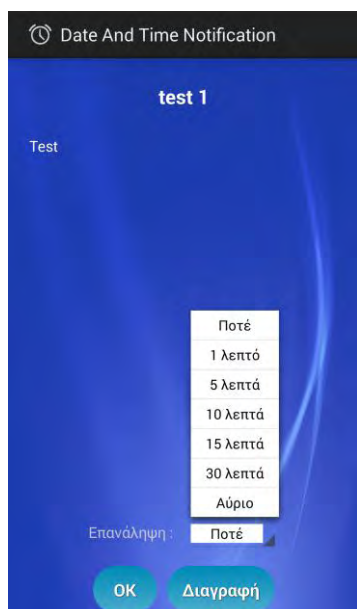
Εικόνα 61. Συγχρονισμένη ειδοποίηση με το Ημερολόγιο



Εικόνα 60. Συγχρονισμένη ειδοποίηση με Google Calendar

## 4.2 Ειδοποίηση χρήστη από την εφαρμογή

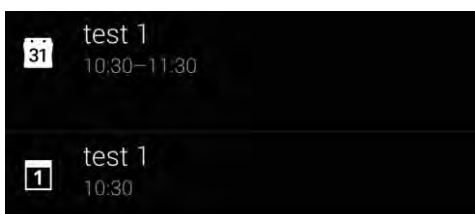
Βασική λειτουργία της εφαρμογής όπως μαρτυρά και το όνομά της είναι η ειδοποίηση του χρήστη όποτε ή όπου έχει αυτός ορίσει. Για την περίπτωση της ειδοποίησης με βάση την ημερομηνία, μόλις φτάσει η επιθυμητή ώρα τότε εμφανίζεται μια ειδοποίηση στην συσκευή με τίτλο, τον τίτλο της ειδοποίησης, και την περιγραφή της, συνοδευόμενη από ένα χαρακτηριστικό ηχητικό μήνυμα και ενεργοποίηση του Led ειδοποίησης. Πατώντας πάνω της ανοίγει μια καρτέλα της εφαρμογής και εμφανίζεται ο τίτλος και η περιγραφή της ειδοποίησης. Ο χρήστης μπορεί να την διαγράψει, να την αφήσει αποθηκευμένη στην εφαρμογή απλά πατώντας το κουμπί OK ή να την ενεργοποιήσει ξανά μετά από κάποιο χρονικό διάστημα που επιλέγεται από μια λίστα που βρίσκεται κάτω από την περιγραφή της. Παρακάτω δίνεται ένα παράδειγμα ειδοποίησης ημερομηνίας.



Εικόνα 65. Ειδοποίηση ημερομηνίας

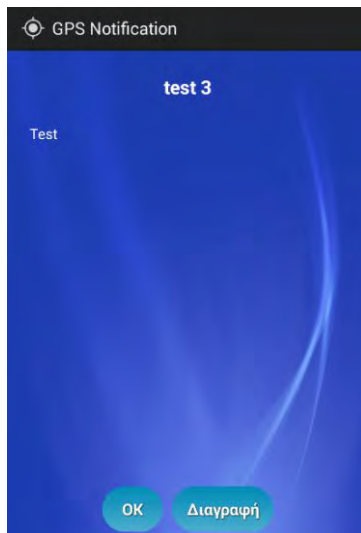


Εικόνα 63. Ειδοποίηση από την εφαρμογή στην μπάρα της συσκευής



Εικόνα 64. Ειδοποίηση από Google Calendar και την εφαρμογή Ημερολόγιο

Για την περίπτωση ειδοποίησης με βάση την τοποθεσία, όταν ο χρήστης φτάσει σε μια απόσταση από το σημείο ειδοποίησης, που εξαρτάται από την επιλογή που έχει στις ρυθμίσεις με default τα 10 μέτρα (Κανονική απόσταση), τότε όπως και στην ειδοποίηση ημερομηνίας εμφανίζεται η ειδοποίηση στην συσκευή. Η μόνη διαφορά με τον άλλο τύπο στην καρτέλα ειδοποίησης που ανοίγει είναι η έλλειψη της επιλογής επανάληψης. Για να ενεργοποιηθεί ξανά πρέπει να πατήσει ο χρήστης Επεξεργασία και μετά πάλι Αποθήκευση. Παρακάτω δίνεται ένα παράδειγμα ειδοποίησης τοποθεσίας.




Εικόνα 67. Ειδοποίηση τοποθεσίας



Εικόνα 66. Ειδοποίηση τοποθεσίας από την εφαρμογή στην μπάρα της συσκευής

### 4.3 Ρυθμίσεις εφαρμογής

Η εφαρμογή επιτρέπει στον χρήστη να κάνει και κάποιες ρυθμίσεις μέσα από την αντίστοιχη καρτέλα. Για να μεταβεί εκεί πρέπει να πατήσει το menu button του Android ή το  αν δεν υπάρχουν κουμπιά στην συσκευή.

Μέσα από τις ρυθμίσεις ο χρήστης μπορεί να ενεργοποιήσει – απενεργοποιήσει την μουσική που παίζει κατά την εκκίνηση της εφαρμογής. Επίσης μπορεί να ορίσει την επιθυμητή απόσταση από το σημείο ειδοποίησης, που αφορά τις ειδοποιήσεις τοποθεσίας και μόνο, επιλέγοντας ανάμεσα σε τρία επίπεδα με:

1. **Κανονικό** περίπου 10 μέτρα από το σημείο και λιγότερο
2. **Ακριβής** περίπου 3 μέτρα και λιγότερο
3. **Λίγο Μακριά** περίπου 15 μέτρα και λιγότερο

Με βάση την παραπάνω επιλογή το GPS “κοιτάει” συνεχώς την απόσταση από το σημείο και μόλις φτάσει τα μέτρα που έχει επιλέξει ο χρήστης γίνεται η ειδοποίηση. Από default είναι επιλεγμένο το Κανονικό.

Τέλος υπάρχει η επιλογή Reset. Πατώντας την ο χρήστης διαγράφει όλες τις ειδοποιήσεις που υπάρχουν στην μνήμη της εφαρμογής και μηδενίζεται η βάση δεδομένων.

## 5. Τελικές Σκέψεις - Επίλογος

Στην παρούσα εφαρμογή χρησιμοποιήθηκαν αρκετά στοιχεία που προσφέρει το Google API με σκοπό την υλοποίηση της εφαρμογής. Έτσι γνώρισα αρκετά καλά το περιβάλλον και τις δυνατότητές του. Μπορώ να πω ότι είναι ένα ιδιαίτερα όμορφο αντικείμενο να ασχοληθεί κανείς.

Ελπίζω η εφαρμογή να είναι εύκολη στην χρήση και χρήσιμη για πολλούς που έχουν θέματα με το να θυμούνται πράγματα. Αυτός ήταν και ο σκοπός μου όταν την σκέφτηκα, να βοηθήσω εμένα και συνάμα όλα τα άτομα που ξεχνάνε εύκολα. Από εκεί και πέρα προσπάθησα μέσα στο εξάμηνο να την εξελίξω όσο πιο πολύ μπορούσα και να προσθέτω συνέχεια νέες λειτουργίες σε συνεργασία βέβαια με τον κύριο Ακρίτα.

Θα με χαροποιούσε ιδιαίτερα το γεγονός να ακούσω σχόλια και προτάσεις από συμφοιτητές μου, αλλά και να συνεργαστώ με οποιονδήποτε θέλει να ασχοληθεί είτε με την επέκταση της εφαρμογής είτε με τη δημιουργία μιας καινούριας. Κάθε σκέψη δικιά σας είναι ευπρόσδεκτη και μπορείτε να την στείλετε μέσω email από την καρτέλα About της εφαρμογής. Όσοι λοιπόν βρήκατε ενδιαφέρον διαβάζοντας τις παραπάνω σελίδες ή παίζοντας με την εφαρμογή στο κινητό σας μη διστάσετε να επικοινωνήσετε μαζί μου.

Στο μέλλον έχω σκοπό να την εξελίξω και να προσθέσω και άλλες, νέες λειτουργίες. Πρωταρχικός βέβαια στόχος είναι να την ανεβάσω στο Google Play Store. Σαν τελικό συμπέρασμα λοιπόν, έχω να δηλώσω, ότι το Android είναι κάτι που σίγουρα θέλω να ασχοληθώ και στο μέλλον καθώς το περιβάλλον και αυτά που προσφέρει είναι πολύ ενδιαφέρον.





## 6. Βιβλιογραφία

1. Στοιχεία σχετικά με την αρχιτεκτονική του Android  
[http://www.tutorialspoint.com/android/android\\_architecture.htm](http://www.tutorialspoint.com/android/android_architecture.htm)
2. Beginning Android Application Development, Wei – Meng Lee
3. Ιστορία και εκδόσεις του Android  
[http://en.wikipedia.org/wiki/Android\\_%28operating\\_system%29](http://en.wikipedia.org/wiki/Android_%28operating_system%29)  
<http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/>
4. Σχετικά με την ανάπτυξη κώδικα  
<http://developer.android.com>
5. Android application development tutorials  
<http://www.mybringback.com/series/android-basics>