

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ
ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη εφαρμογής αξιοποίησης των αισθητήρων κινητού
τηλεφώνου για το λειτουργικό σύστημα Android**

**Development of an Android application that utilizes a mobile phone's
sensors**



Επιμέλεια: Ζαχαριάδου Ευαγγελία
Μαγκούτα Αγνή

Επιβλέπων: Στάμος Κωνσταντίνος

ΠΑΤΡΑ 2014

Περίληψη

Με την πρόοδο της τεχνολογίας τα τελευταία χρόνια, παρατηρείται μια μεγάλη αύξηση στην χρήση των λεγόμενων “smartphones”, δηλαδή “έξυπνων κινητών τηλεφώνων”. Τα κινητά τηλέφωνα πλέον έχουν να προσφέρουν πολλά περισσότερα από την άμεση επικοινωνία. Παρέχουν τη δυνατότητα επικοινωνίας με όλο τον κόσμο μέσω του ίντερνετ, ψυχαγωγία με παιχνίδια που μπορεί ο καθένας να διαλέξει και να κατεβάσει στο κινητό του στη στιγμή, ενημέρωση με online εφημερίδες, συντροφιά ηλεκτρονικών βιβλίων, γρήγορη πρόσβαση σε σελίδες κοινωνικής δικτύωσης, δυνατότητα λήψης και αποστολής φωτογραφιών και πολλές άλλες ιδιαιτερότητες.

Ξαφνικά, ένας ολοκαίνουριος κόσμος έγινε διαθέσιμος και προσιτός οικονομικά στο ευρύ κοινό, παρέχοντας πρόσβαση παντού, απ’ όπου κι αν βρίσκεται, ενώ μέχρι πρότινος οι περισσότερες δυνατότητες ήταν εγκλωβισμένες μόνο στους ηλεκτρονικούς υπολογιστές. Τώρα πλέον τα κινητά τηλέφωνα αναπτύσσονται σε ένα πρότυπο “όλα-σε-ένα”, δηλαδή υπηρεσίες πολλών συσκευών συγκεντρωμένες σε μία. Ένα μεγάλο άλμα στη χρηστικότητα των smartphones προσέφερε η πλατφόρμα της Google, Android.

Σε αυτή την πτυχιακή εργασία αναπτύχθηκε μια εφαρμογή κινητού τηλεφώνου με χρήση της πλατφόρμας Google Android, σε συνεργασία με το λογισμικό ανοιχτού κώδικα Eclipse, το οποίο συνεργάζεται με τα εργαλεία ανάπτυξης Android που προσφέρονται δωρεάν από την Google (Android Development Tools).

Η εφαρμογή που αναπτύχθηκε έχει σαν σκοπό να εξυπηρετήσει έναν Δήμο να εντοπίσει και να αντιμετωπίσει πιο γρήγορα τις βλάβες και τα προβλήματα που μπορεί να προκύψουν, δίνοντας στους πολίτες έναν γρήγορο τρόπο να ενημερώνουν το Δήμο μέσω της εφαρμογής, στέλνοντας ένα αίτημα με την ακριβή τοποθεσία, μια φωτογραφία του προβλήματος και προαιρετικά και κάποιες σημειώσεις.

Έτσι, οι πολίτες παίρνουν έναν πιο ενεργό ρόλο τονίζοντας τα προβλήματα στην πόλη που ζουν και ο Δήμος μπορεί να ανταπεξέλθει πιο γρήγορα και στοχευμένα. Ο κύριος στόχος της πτυχιακής εργασίας ήταν η ανάπτυξη μιας εφαρμογής κινητού τηλεφώνου με τη χρήση του λογισμικού Android για την ανάδειξη των αισθητήρων, παρατηρώντας έτσι το πώς η εκμετάλλευσή τους μπορεί να βελτιώσει τη χρηστικότητα ενός smartphone.

Abstract

Technological progress led to massively increased usage of the so-called smartphones. This new era in mobile communications took the next step by offering much more than simple communication tools. They opened a whole new world of entertainment and simultaneously managed to improve upon the ways we connect to each other, via the internet. Everything a person wanted, be it books, music, movies or simply the daily newspaper, came to be just a few finger moves away. On top of that, smartphones dethroned digital cameras offering viable picture quality combined with immediate access to uploading and sharing.

Even though the cost of a decent smartphone was quite steep at first, the prices declined rapidly. Suddenly the whole world was available and open to middle class people and below. Things that could only be accomplished by sitting in front of a computer can now be done on the move, using smartphones. This new all-in-one model led to a huge technological leap that managed to change people's lives. Android operating system, developed by Google, is one of the driving forces that contributed to it.

The main subject of this thesis is the development of a mobile application for phones based on the Android platform. Eclipse, an open source software, was used for this task along with Android Development Tools, a set of free to use development tools by Google.

The basic premise that led to Citizen Call was that of providing the means to simplify the ability of any citizen to inform the pertinent department of his municipality, be it town or city, about everyday situations that occur and demand correction. This could be anything like a road part collapsing or a flood. That way the citizens actively contribute to the rectification of these problems by offering a fast and direct stream of information to the authorities, making their work easier. The main goal of this thesis was the development of an application by using the Android platform in conjunction with the various sensors a modern smartphone includes, observing and recording all the while how this improves the phones usability by the user.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	1
ABSTRACT	2
ΕΙΣΑΓΩΓΗ	6
ΚΕΦΑΛΑΙΟ 1: Κινητές συσκευές	
1.1 Εισαγωγή	8
1.2 Τι είναι το κινητό τηλέφωνο	8
1.3 Ιστορική αναδρομή κινητών τηλεφώνων	9
1.4 Τι είναι το smartphone	17
1.5 Κυψελοειδές Δίκτυο	17
1.5.1 Λειτουργία Σταθμού Βάσης	18
1.5.2 Διάσπαση Κυψελών	19
1.6 GSM	20
1.7 Δομή του GSM	20
1.7.1 Κινητός Σταθμός	21
1.7.2 Υποσύστημα Σταθμού Βάσης	21
1.7.3 Υποσύστημα Δικτύου και Μεταγωγής	22
1.7.4 Υποσύστημα Λειτουργίας και Υποστήριξης	23
1.8 Ορισμός ενός αισθητήρα	24
1.9 Αισθητήρες σε κινητά τηλέφωνα	24
1.9.1 Οθόνη Αφής	24
1.9.2 Μικρόφωνο	25
1.9.3 Ψηφιακή Φωτογραφική Μηχανή (Κάμερα)	25
1.9.4 GPS	25
1.9.5 Επιταχυνσιόμετρο	27
1.9.6 Γυροσκόπιο	28
1.9.7 Αισθητήρας Φωτός	29
1.9.8 Εγγύτητα	29
1.10 Tablets	29
ΚΕΦΑΛΑΙΟ 2: Λειτουργικά Συστήματα Κινητών	
2.1 Εισαγωγή	31
2.2 Ορισμός λειτουργικού Συστήματος	31

2.3	Τι είναι το Android	31
2.4	Ιστορική Αναδρομή	32
2.5	Ορισμός του API	33
2.6	Αρχιτεκτονική του Android	34
2.6.1	Linux Kernel	35
2.6.2	Libraries	35
2.6.3	Android Runtime	36
2.6.4	Application Framework	36
2.6.5	Applications	38
2.7	Εκδόσεις Android	39
2.8	Ποσοστά χρήσης των εκδόσεων Android	52
2.9	Επιλογή έκδοσης για την εφαρμογή	54
2.10	Άλλα λειτουργικά συστήματα κινητών συσκευών	54
2.10.1	iOS	54
2.10.2	Blackberry OS	54
2.10.3	Windows Mobile	55
2.10.4	Symbian	55
ΚΕΦΑΛΑΙΟ 3: Σχεδιασμός και Ανάλυση Εφαρμογής		
3.1	Εισαγωγή	57
3.2	Αρχική ιδέα μιας εφαρμογής	57
3.3	Στόχος της εφαρμογής	57
3.4	Εγκατάσταση λογισμικού	58
3.4.1	Android SDK	58
3.4.2	Eclipse	63
3.5	Το εσωτερικό μιας εφαρμογής	66
3.6	Ασφάλεια εφαρμογών Android	67
3.7	Συστατικά εφαρμογής	68
3.8	Το αρχείο Manifest	70
3.9	Σχεδιασμός της εφαρμογής	70
3.9.1	Καθορισμός των Δραστηριοτήτων (Activities)	72
3.9.2	Δήλωση των Δραστηριοτήτων στο Manifest	74
3.9.3	Δημιουργία Layout	76
3.9.4	Δημιουργία Layout των Μενού	79

3.10 Προετοιμασία εικόνων και κειμένου	80
3.10.1 Η χρήση των string στο Android	80
3.10.2 Οι φάκελοι γραφικών Drawable	80
ΚΕΦΑΛΑΙΟ 4: Υλοποίηση Εφαρμογής	
4.1 Εισαγωγή	82
4.2 Υλοποίηση της Αρχικής Οθόνης	82
4.2.1 Το layout_main.xml	82
4.2.2 Η κλάση Main.java	84
4.2.3 Το Μενού About	85
4.3 Υλοποίηση της Οθόνης Έναρξης	87
4.3.1 Το layout start.xml	88
4.3.2 Η κλάση startscreen.java	90
4.3.3 Δήλωση αδειών στο Manifest	95
4.3.4 Το Μενού Ρυθμίσεων και Βοήθειας	96
4.4 Υλοποίηση της Οθόνης Βοήθειας	97
4.4.1 Το layout help.xml	97
4.4.2 Η κλάση helpscreen.java	98
4.5 Δοκιμές λειτουργίας	98
4.6 Debugging	99
4.6.1 LogCat	99
4.6.2 Dalvik Debug Monitor Server (DDMS)	100
4.7 Τελική έκδοση	101
ΣΥΜΠΕΡΑΣΜΑΤΑ	103
ΒΙΒΛΙΟΓΡΑΦΙΑ	104

Εισαγωγή

Οι σύγχρονες συσκευές κινητής τηλεφωνίας αποτελούν υπερπλήρη υπολογιστικά συστήματα τόσο λόγω επεξεργαστικής ισχύς όσο και λόγω διαδραστικών δυνατοτήτων. Ενσωματώνοντας μια ευρεία γκάμα αισθητήρων εικόνας και ήχου τείνουν να αντικαταστήσουν πολλές εξειδικευμένες συσκευές όπως φωτογραφικές μηχανές (ειδικά μικρής και μεσαίας κλάσης), GPS, δημοσιογραφικά κασετόφωνα ηχογράφησης και άλλα.

Κινητήρια σκέψη και πηγή έμπνευσης για την εκπόνηση της παρούσης εφαρμογής στάθηκε η συλλογική και πολύπλευρη διάσταση που αποκτά μια συσκευή όταν διαθέτει τέτοια πλειάδα αισθητήρων. Όταν κάθε μια από αυτές τις μεμονωμένες συσκευές δύναται να εξυπηρετήσει ορισμένες ανάγκες της ανθρώπινης δραστηριότητας σε πολυεπίπεδη βάση, αποκτά μεγάλο ενδιαφέρον η εξερεύνηση του οφέλους που μπορεί κανείς να αποκομίσει με την συνεργιστική τους δράση. Προς εξυπηρέτηση λοιπόν της καθημερινής ζωής του πολίτη, με αφετηρία την συλλογιστική προσέγγιση του κινητού τηλεφώνου ως προέκταση των φυσικών αισθητήρων του ανθρώπινου σώματος, επιλέχθηκε η δημιουργία του Citizen Call. Με την χρήση του ο πολίτης μπορεί να μεταδώσει αυτούσιο, επεξηγημένο και τοπικά προσδιορισμένο με σαφήνεια το όποιο πρόβλημα παρατηρεί στους δρόμους της πόλης του, κατευθείαν στις αρμόδιες αρχές.

Η κύρια πρόκληση κατά τον προγραμματισμό και την υλοποίηση της εφαρμογής ήταν ο σωστός συνδυασμός των αισθητήρων της φωτογραφικής μηχανής του κινητού καθώς και της συσκευής GPS, ώστε να μπορεί να αποσταλεί με την ελάχιστη δυνατή μεσολάβηση του χρήστη η απεικόνιση του εκάστοτε προβλήματος που παρουσιάζεται σε κάποιο σημείο της πόλης μαζί με το ακριβές στίγμα, ώστε να καθίσταται ευκόλως εντοπίσιμο από τις αρχές.

Σκόπιμο κρίθηκε να υπάρχουν μηνύματα ενημέρωσης για την περίπτωση που το υποσύστημα του GPS παρέμενε κλειστό, κάτι που εξάλλου συνηθίζεται από αρκετούς χρήστες κινητών τηλεφώνων για εξοικονόμηση ενέργειας, ώστε να ενεργοποιηθεί προτού πραγματοποιηθεί η χρήση της εφαρμογής.

Το γραφικό περιβάλλον της εφαρμογής κρίθηκε σκόπιμο να κρατηθεί όσο γίνεται περισσότερο λιτό και απλό ώστε να προαχθεί η αμεσότητα και ευκολία της χρήσης του. Οποιαδήποτε προσπάθεια για κάτι παραπάνω από αυτό πιο πολύ θα

παρεμπόδιζε την ξεκάθαρη εικόνα που παρέχεται στον χρήστη και δεν θα προσέφερε τίποτα ουσιαστικό σε πρακτικό επίπεδο.

Στο πρώτο κεφάλαιο γίνεται αρχικά μια εισαγωγή στις συσκευές κινητών τηλεφώνων, τα χαρακτηριστικά, τον τρόπο λειτουργίας τους καθώς και τους αισθητήρες που περιέχουν. Ακολουθεί το κεφάλαιο των λειτουργικών συστημάτων που είναι απαραίτητα για τη λειτουργία μιας υπολογιστικής συσκευής. Τέλος, το σημαντικότερο κομμάτι της εργασίας αποτελεί ο σχεδιασμός και η υλοποίηση της εφαρμογής που παρουσιάζεται στα τελευταία δυο κεφάλαια. Γίνεται μια εκτενής παρουσίαση του κώδικα και των λογισμικών που έχουν χρησιμοποιηθεί για να δημιουργηθεί η εφαρμογή Citizen Call.

ΚΕΦΑΛΑΙΟ 1

Κινητές Συσκευές

1.1 Εισαγωγή

Η ανάγκη του ανθρώπου για επικοινωνία και πληροφορία τον ώθησε στο πέρας των χρόνων σε μεγάλες εφευρέσεις. Μια από αυτές ήταν το σταθερό τηλέφωνο, το οποίο με την χρήση ηλεκτρικών σημάτων έχει τη δυνατότητα να μεταφέρει τον ήχο μεταξύ δυο συσκευών. Μέσα από διάφορες εξελίξεις, το σταθερό τηλέφωνο έγινε ασύρματο για ελευθερία κίνησης μέσα στο σπίτι και στη συνέχεια έκανε την εμφάνισή του το κινητό τηλέφωνο, το οποίο έδωσε στους ανθρώπους τη δυνατότητα επικοινωνίας σε οποιαδήποτε στιγμή, από όπου κι αν βρίσκονται. Η αναπτυσσόμενη τεχνολογία έδωσε περισσότερες ικανότητες στις κινητές συσκευές πέραν της ομιλίας, και συνδύασε τις δυνατότητες διαφόρων συσκευών σε μια, φτάνοντας στο σήμερα όπου πλέον οι κινητές συσκευές προσφέρουν όσα και οι ηλεκτρονικοί υπολογιστές, ενώ έχουν το πλεονέκτημα του μικρότερου μεγέθους.

1.2 Τι είναι το κινητό τηλέφωνο

Το κινητό τηλέφωνο (ή απλά κινητό) είναι μια τηλεφωνική συσκευή, η οποία χρησιμοποιεί ένα ασύρματο δίκτυο για να πραγματοποιήσει ή να λάβει κλήσεις. Σε αντίθεση με το σταθερό τηλέφωνο, δεν εξαρτάται από καλώδια για τη σύνδεσή του με τον εκάστοτε πάροχο. Με αυτό τον τρόπο, ο χρήστης έχει τη δυνατότητα επικοινωνίας οπουδήποτε κι αν βρίσκεται.

Τα εξαρτήματα που συναντώνται σε όλα τα κινητά τηλέφωνα είναι:

- Η οθόνη
- Η μπαταρία
- Το πληκτρολόγιο ή η οθόνη αφής, για να μπορεί να αλληλοεπιδρά ο χρήστης με το τηλέφωνο
- Αισθητήρες, όπως μικρόφωνο και μεγάφωνο για τις κλήσεις
- Μια υποδοχή για την κάρτα SIM
- Η πλακέτα που στηρίζεται όλο το hardware, τα κυκλώματα, καθώς και την κεραία του κινητού
- Ο εξωτερικός σκελετός, το κάλυμμα όλων των παραπάνω

1.3 Ιστορική αναδρομή κινητών τηλεφώνων

Οι περισσότεροι άνθρωποι όταν καλούνται να θυμηθούν τα παλιά κινητά τηλέφωνα φέρνουν στο νου τους ογκώδεις συσκευές με μεγάλες κεραίες. Στην πραγματικότητα, το πρώτο κινητό τηλέφωνο, από την άποψη της ελευθερίας κίνησης, εμφανίστηκε αρχικά για αυτοκίνητα το 1946 στο Σαίντ Λούις των Ηνωμένων Πολιτειών. Ο εξοπλισμός ήταν πιο ακριβός από ένα καινούριο αυτοκίνητο και ζύγιζε 36 κιλά (εικόνα 1.1).



Εικόνα 1.1: Το κινητό τηλέφωνο για αυτοκίνητα, η βάση του και ο εξοπλισμός.
Πηγή <http://weburbanist.com>, Άρθρο *Remember millions of mobile phones in the 1960s? You should*

Ο τρόπος λειτουργίας του θυμίζει αρκετά αυτό των ραδιοφώνων. Χρειαζόταν ένας κύριος πομποδέκτης με μια κεραία και υπήρχαν διαθέσιμα κανάλια για την πραγματοποίηση των κλήσεων σε συγκεκριμένες ραδιοσυχνότητες. Οι κλήσεις δεν γινόταν απευθείας καθώς ο οδηγός έπρεπε πρώτα να συνδεθεί με το κεντρικό σταθμό και να δώσει στον διαχειριστή τον τηλεφωνικό αριθμό, και έπειτα αυτός πραγματοποιούσε την κλήση. Η ποιότητα των κλήσεων ήταν κακή, οι κλήσεις κοβόταν συχνά, ενώ αν όλα τα διαθέσιμα κανάλια ήταν κατειλημμένα, ο καλών έπρεπε να περιμένει να ελευθερωθεί κάποιο. Η χρήση του θυμίζει αυτή των walkie-talkie, καθώς για να μιλήσει ο χρήστης έπρεπε να κρατάει πατημένο ένα κουμπί και να το ελευθερώνει για να ακούσει. Ο εξοπλισμός αποτελείτο από πομπούς και δέκτες ραδιοσυχνοτήτων χαμηλής ισχύος και μια μπαταρία. Τα τηλέφωνα στο αυτοκίνητο γίνονται όλο και πιο διαδεδομένα, μέχρι που το 1973, η Motorola παρήγαγε το πρώτο κινητό τηλέφωνο χειρός (εικόνα 1.2). Το βάρος του ήταν περίπου 1 κιλό και οι διαστάσεις του κάθε άλλο παρά μικρές, 23 εκατοστά μήκος, 13 εκατοστά βάθος και 4.45 εκατοστά πλάτος. Η μπαταρία του αρκούσε μόνο για 30 λεπτά ομιλίας και έπειτα χρειαζόταν 10 ώρες μέχρι να φορτίσει.



Εικόνα 1.2: Το πρώτο κινητό τηλέφωνο.

Πηγή: <http://www.glbrain.com>, Άρθρο *First cell phone! DynaTAC! Motorola!*

Ο σχεδιαστής του, Μάρτιν Κούπερ, έκανε την πρώτη κλήση περπατώντας στη Νέα Υόρκη προς τον ανταγωνιστή του, Τζόελ Ενγκελ. Το 1976 υπήρχαν μόλις 545 κάτοχοι κινητού τηλεφώνου στην Νέα Υόρκη, ενώ παράλληλα χιλιάδες ήταν σε λίστα αναμονής. Το πρόβλημα των παρόχων ήταν το περιορισμένο εύρος ζώνης (bandwidth), το φάσμα δηλαδή των συχνοτήτων μέσα σε ένα προκαθορισμένη ζώνη που χρησιμοποιούνται για τη μετάδοση ενός σήματος. Η λύση σε αυτό το πρόβλημα ήταν η χρήση κυψελοειδούς συστήματος (cellular system). Το πρώτο εμπορικά διαθέσιμο αυτοματοποιημένο κυψελοειδές δίκτυο ήταν της εταιρείας Nippon Telegraph and Telephone Corporation (NTT) στην Ιαπωνία, το 1979. Ακολούθησε η Nordic Mobile Telephone (NMT) το 1981 στη Δανία, τη Νορβηγία, τη Σουηδία και τη Φινλανδία και μέσα στη δεκαετία αυτή ακολούθησαν και άλλες χώρες. Αυτή ήταν η πρώτη γενιά συστημάτων (1G) και ήταν αναλογική.

Το πρώτο εμπορικά διαθέσιμο τηλέφωνο κυκλοφόρησε στις 21 Σεπτεμβρίου 1983 και ήταν το μοντέλο DynaTAC της Motorola (εικόνα 1.3).



Εικόνα 1.3: Το Motorola DynaTAC.

Πηγή: <http://dreisechsnnull.telekom.de>, Άρθρο *30 Jahre Hande*

Τα κινητά τηλέφωνα έγιναν δημοφιλή στις Ηνωμένες Πολιτείες με εκτιμήσεις της τάξεως των 340,213 πελατών το 1985 οι οποίοι ανήλθαν σε 1,300,855 μέχρι το 1987.

Στην Ευρώπη προέκυψαν τα πρώτα προβλήματα, καθώς δεν υπήρχε η δυνατότητα χρήσης του κινητού τηλεφώνου πέρα από τα σύνορα της χώρας του κατόχου. Έτσι, δημιουργήθηκε ένα νέο σύστημα δευτέρας γενιάς (2G), τα ψηφιακά κυψελοειδή δίκτυα με την ονομασία Groupe Speciale Mobile (GSM). Πλέον τα αρχικά σημαίνουν Global System for Mobile Communications.

Η πρώτη κινητή συσκευή "τσέπης" ήταν το MicroTAC της Motorola (εικόνα 1.4), το οποίο κυκλοφόρησε το 1989. Ήταν η πρώτη συσκευή με καπάκι, το μήκος της οποίας ήταν 22 εκατοστά όταν το καπάκι ήταν ανοιχτό.



Εικόνα 1.4: Το Motorola MicroTAC.

Πηγή: <http://en.wikipedia.org>, Άρθρο *Motorola MicroTAC*

Το γραπτό μήνυμα ή απλώς μήνυμα (Short Message Service - SMS), είναι μια από τις 2 πιο δημοφιλείς λειτουργίες ενός κινητού τηλεφώνου. Το πρώτο μήνυμα στάλθηκε στις 3 Δεκεμβρίου 1992 στην Αγγλία από τον Neil Papworth. Η υπηρεσία αποστολής και λήψης γραπτών μηνυμάτων έγινε αρχικά διαθέσιμη το 1993 στη Φινλανδία, σε συσκευές που μπορούσαν να την υποστηρίξουν. Η πρώτη εταιρεία που παρείχε τέτοιες συσκευές ήταν η Nokia. Ένα τυπικό μήνυμα έχει μέγεθος 140 bytes, τα οποία μεταφράζονται σε 160 λατινικούς χαρακτήρες (70 όταν υπάρχει κωδικοποίηση για άλλες γλώσσες με μη-λατινικούς χαρακτήρες). Μέχρι το 2010 είχαν σταλεί 6,1 τρισεκατομμύρια SMS.

Ο πρόγονος των smartphone έρχεται το 1993 και ονομάζεται Simon Personal Communicator, της εταιρείας IBM (εικόνα 1.5). Παρέχει υπηρεσίες αποστολής fax,

e-mail, ανάγνωσης σελίδων στο internet, ενώ περιέχει τηλεφωνικό κατάλογο, ημερολόγιο, σημειωματάριο, αριθμομηχανή, ρολόι και οθόνη αφής με πενάκι.



Εικόνα 1.5: Το πρώτο smartphone, το Simon της IBM.

Πηγή: <http://en.wikipedia.org>, Άρθρο *IBM_SIMON*

Μέσα στην δεκαετία του 1990, τα κινητά τηλέφωνα άρχισαν να αποκτούν όλο και περισσότερες λειτουργίες, ενώ συρρικνώνονταν σε μέγεθος. Το 1998 κυκλοφόρησε το 8810 της Nokia, το πρώτο κινητό τηλέφωνο με εσωτερική κεραία για αισθητικούς λόγους. Το 1999 ήταν η χρονιά που έκανε την εμφάνισή του το Nokia 3210, ένα από τα δημοφιλέστερα κινητά τηλέφωνα στον κόσμο με πωλήσεις που ανέρχονται στα 160 εκατομμύρια συσκευές. Είχε τη δυνατότητα αποστολής προεγκατεστημένων εικόνων σε μηνύματα, περιείχε 3 παιχνίδια και ένα πρόγραμμα δημιουργίας ήχων κλήσης (ringtones). Την ίδια χρονιά κυκλοφόρησε το Nokia 7110, το πρώτο κινητό τηλέφωνο με Wireless Application Protocol

(WAP), ένα τεχνικό πρότυπο το οποίο παρέχει τη δυνατότητα πρόσβασης στο internet.

Το Benefon Esc! ήταν το πρώτο κινητό τηλέφωνο με ενσωματωμένο GPS (εικόνα 1.6) και ελαφριά αδιάβροχη προστασία. Ανακοινώθηκε το 1999 και κυκλοφόρησε το 2000.



Εικόνα 1.6: Το Benefon Esc!, το πρώτο κινητο με GPS.

Πηγή: <http://freemarket.kiev.ua>, Άρθρο *Benefon Esc*

Στις αρχές της δεκαετίας του 2000, η τεχνολογία των κινητών συσκευών προχωράει ταχύτατα και μια σειρά νέων δυνατοτήτων εμφανίζονται σε όλο και περισσότερα μοντέλα. Τα τηλέφωνα αρχίζουν και υποστηρίζουν αρχεία μουσικής mp3, έχουν ενσωματωμένη φωτογραφική μηχανή, έχουν bluetooth, υπέρυθρες ακτίνες, ραδιόφωνο, εσωτερική μνήμη και η λίστα συνεχίζεται. Οι πρώτες έγχρωμες οθόνες ήταν γεγονός, ενώ οι κατασκευαστές έδιναν μεγάλη σημασία στην εξωτερική εμφάνιση και το μέγεθος μιας συσκευής. Μετά το 2005 κάνουν την εμφάνισή τους συσκευές τρίτης γενιάς (3G).

Το 2007, ο Steve Jobs και η Apple παρουσιάζουν το iPhone (εικόνα 1.7), μια πρωτοπόρα συσκευή με οθόνη αφής και αισθητήρες για αυτόματη περιστροφή οθόνης ή πολλαπλά αγγίγματα σε αυτήν.



Εικόνα 1.7: Το πρώτο iPhone.

Πηγή: <http://appadvice.com>, Άρθρο *iPhone 2G Loaded With iOS 4 Features*

Την επόμενη χρονιά, κυκλοφορεί η πρώτη συσκευή με λογισμικό Android, HTC Dream. Μέχρι το τέλος της δεκαετίας του 2000, οι συσκευές κινητών τηλεφώνων πλέον συμπεριλαμβάνουν δυνατότητες πολλών ηλεκτρονικών συσκευών σε μια. Κινητά με οθόνη αφής αρχίζουν και αντικαθιστούν τα πληκτρολόγια, οι κάμερες έχουν καλύτερη ανάλυση κάθε χρόνο και η πρόσβαση στο internet γίνεται πλέον μέσω GPRS και WiFi.

Μετά το 2010, ο κόσμος άρχισε να τείνει περισσότερο προς τα smartphones. Οι συσκευές άρχισαν να μεγαλώνουν πάλι σε μέγεθος, προκειμένου η οθόνη αφής να καλύπτει μεγαλύτερη έκταση. Οι συσκευές άρχισαν να γίνονται πιο φιλικές με το internet, άρχισαν να περιέχουν εφαρμογές για σελίδες κοινωνικής δικτύωσης, περισσότερα παιχνίδια, καλύτερη ποιότητα εικόνας, κάμερες με περισσότερα megapixel, γρηγορότερους επεξεργαστές, περισσότερους αισθητήρες και συνεχίζουν την εξέλιξή τους σε μηχανήματα όλα-σε-ένα. Τα smartphones του σήμερα επιτρέπουν στους χρήστες να πλοηγούνται στο internet με ευχέρεια, να

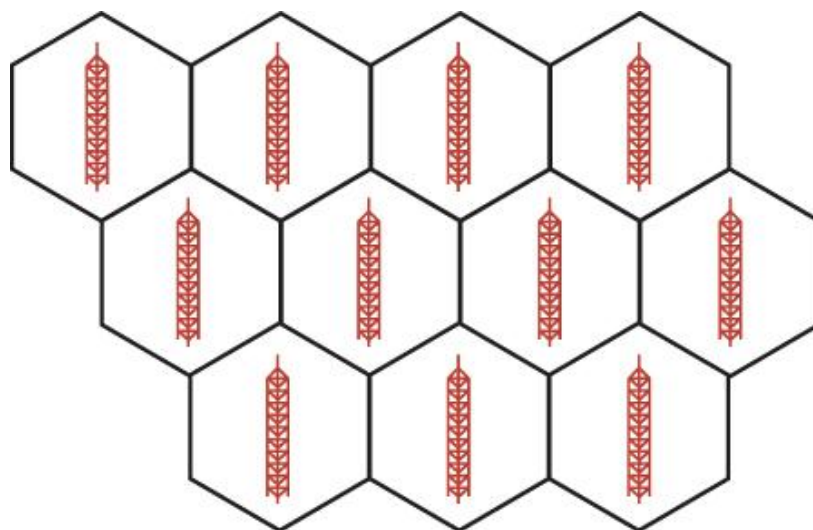
επικοινωνούν όχι μόνο τηλεφωνικά αλλά και με εικόνα, ενώ οι δυνατότητές τους θυμίζουν περισσότερο υπολογιστές τσέπης.

1.4 Τι είναι το smartphone

Smartphone ονομάζεται μια συσκευή κινητού τηλεφώνου, η οποία περιέχει πιο προηγμένες υπολογιστικές δυνατότητες και συνδεσιμότητα σε σχέση με τα βασικά χαρακτηριστικά των κινητών. Συνήθως, ένα smartphone συνδυάζει τα χαρακτηριστικά ενός κινητού, μιας φωτογραφικής μηχανής, μιας κάμερας, μιας συσκευής αναπαραγωγής πολυμέσων, ενός GPS και πολλών άλλων. Όλα αυτά δεν επηρεάζουν το μέγεθος της συσκευής, που μπορεί να κυμαίνεται σε διάφορα μεγέθη ανάλογα με την οθόνη της.

1.5 Κυψελοειδές Δίκτυο

Το Κυψελοειδές Δίκτυο (Cellular Network) είναι ένα ασύρματο δίκτυο υψηλών ραδιοσυχνοτήτων που χρησιμοποιείται από τους παρόχους κινητής τηλεφωνίας. Το μοτίβο που ακολουθεί μοιάζει με αυτό των κυψελών που φτιάχνουν οι μέλισσες, εξού και το όνομά του. Για τη λειτουργία του, αρχικά χωρίζεται η ευρύτερη γεωγραφική περιοχή όπου είναι επιθυμητή η κάλυψη δικτύου σε μικρότερες περιοχές που ονομάζονται κυψέλες. Μέσα σε κάθε κυψέλη υπάρχει ένας πομποδέκτης ή Σταθμός Βάσης (Base Station) όπως φαίνεται στο σχήμα 1.1. Αυτό το πρότυπο επαναλαμβάνεται ωσότου να καλυφθεί η περιοχή.

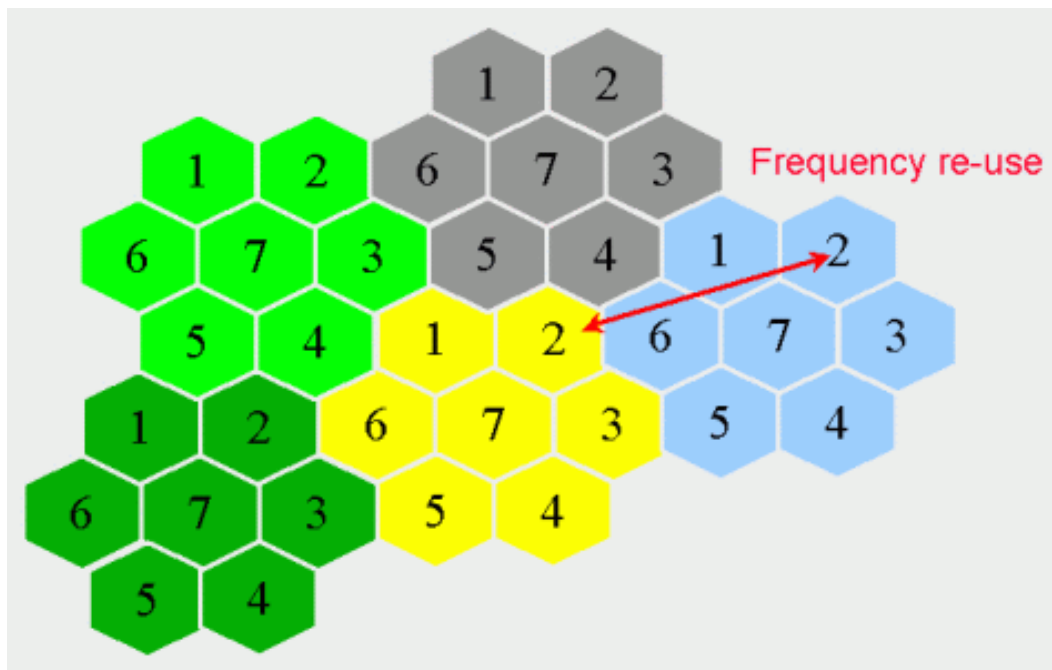


Σχήμα 1.1: Ένα κυψελοειδές δίκτυο.

Πηγή: <http://www.quepublishing.com>, Άρθρο *How Mobile Networks Work*

1.5.1 Λειτουργία σταθμού βάσης

Ο σταθμός βάσης εκπέμπει σε μια συγκεκριμένη συχνότητα, με μικρή ισχύ. Μια κυψέλη έχει διαφορετική συχνότητα με τις γειτονικές της, για την αποφυγή παρεμβολών. Αυτός είναι και ο λόγος της μικρής ισχύος του σταθμού βάσης, ώστε να καλύπτει επαρκώς μόνο το εμβαδόν της κυψέλης που βρίσκεται (και λίγο παραπάνω, ώστε να μην υπάρχουν κενές ζώνες ανάμεσα σε 2 κυψέλες). Ο αριθμός των διαφορετικών συχνοτήτων είναι περιορισμένος, καθώς η κινητή τηλεφωνία χρησιμοποιεί ένα προκαθορισμένο φάσμα συχνοτήτων. Το σύνολο των διαφορετικών συχνοτήτων εμφανίζεται μια συστάδα κυψελών, της οποίας το μοτίβο επαναλαμβάνεται για τη δημιουργία ενός ευρύτερου δικτύου όπως φαίνεται στο σχήμα 1.2. Οι όμοιες συχνότητες έχουν επαρκή απόσταση μεταξύ τους ώστε να αποφευχθούν οι παρεμβολές.



Σχήμα 1.2: Παράδειγμα χρήσης 7 συχνοτήτων σε 5 συστάδες κυψελών.

Πηγή: <http://www.raymaps.com>, Άρθρο *Soft Frequency Reuse in LTE*

Ένας κάτοχος κινητού τηλεφώνου χρησιμοποιεί τη συχνότητα πλησιέστερα στο σημείο που βρίσκεται. Όταν μετακινείται, η ισχύς της συχνότητας φθίνει και μιας άλλης αυξάνεται, οπότε συνδέεται αυτόματα σε αυτήν με το δυνατότερο σήμα,

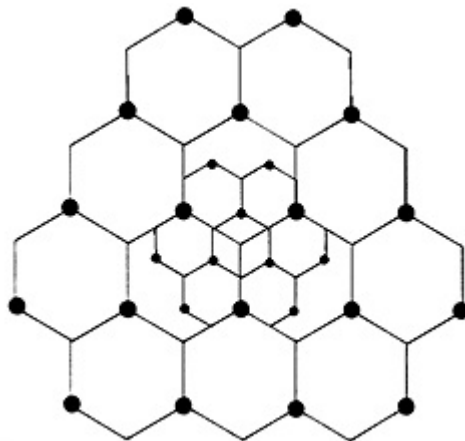
χωρίς να γίνεται αισθητή η μετάβαση από τον χρήστη. Η διαδικασία αυτή ονομάζεται μεταβίβαση (handover).

Ένας σταθμός βάσης τοποθετείται σε σημείο που μπορεί να παρέχει καλύτερη κάλυψη στην εκάστοτε κυψέλη, συνήθως επάνω σε λόφους ή υψωμένα σημεία όπως κτήρια. Όταν ένα κινητό τηλέφωνο είναι ενεργοποιημένο, δέχεται σήματα ελέγχου από τους κοντινούς σταθμούς βάσης. Εφόσον εντοπιστεί ο πλησιέστερος σταθμός βάσης, γίνεται η σύνδεση και αποστέλλονται πληροφορίες για την τοποθεσία του χρήστη εάν αυτός κινείται, για την ανάθεση μιας πιθανής εισερχόμενης κλήσης στην πλησιέστερη σε αυτόν κυψέλη.

1.5.2 Διάσπαση κυψελών

Μια κυψέλη σε μια πυκνοκατοικημένη περιοχή χειρίζεται μεγαλύτερο τηλεπικοινωνιακό φόρτο. Οι συχνότητες της κυψέλης δεν επαρκούν για την εξυπηρέτηση όλων των χρηστών. Η διάσπαση μιας κυψέλης είναι απαραίτητη για να αντιμετωπιστεί αυτό το πρόβλημα.

Μια κυψέλη διασπάται σε μικρότερες με αποτέλεσμα την δημιουργία περισσότερων σταθμών βάσης (σχήμα 1.3). Τα μεγέθη είναι 5, macro, micro, pico, femto και umbrella cells.



Σχήμα 1.3: Διάσπαση κυψελών σε μικρότερες.

Πηγή: <http://www.privateline.com>, Άρθρο *Cellular Telephone Basics*

Ανάλογα με την κίνηση στο δίκτυο, οι μικρότερες κυψέλες μπορούν να ενεργοποιούνται ή να απενεργοποιούνται για αποδοτικότερη χρήση. Όταν ενεργοποιούνται οι μικρότεροι σταθμοί βάσεων, η κεντρική κεραία μειώνει την

ισχύ που εκπέμπει. Έχει παρατηρηθεί ότι η χρήση μικρότερων κυψελών συμβάλει στην ελαχιστοποίηση των παρεμβολών από γειτονικές κυψέλες.

1.6 GSM

Το Global System for Mobile communications, ευρέως γνωστό ως GSM, είναι ένα ψηφιακό σύστημα για τηλεπικοινωνίες κινητής τηλεφωνίας. Είναι το πιο αναπτυγμένο ασύρματο δίκτυο κινητής τηλεφωνίας στον κόσμο. Δίνει στους χρήστες κινητών τη δυνατότητα να χρησιμοποιούν τις συσκευές τους σε όποια χώρα κι αν βρίσκονται χωρίς κανένα πρόβλημα.

Η αρχική ιδέα για τη δημιουργία του προέκυψε καθώς τα κινητά τηλέφωνα στην Ευρώπη δεν μπορούσαν να λειτουργήσουν σε άλλες χώρες (δεν υπήρχε roaming) λόγω των αναλογικών δικτύων κινητής. Με την αύξηση των ταξιδιωτών από χώρα σε χώρα, το πρόβλημα αυτό θα διογκωνόταν εάν δεν αντιμετωπιζόταν από νωρίς. Η δημιουργία του προήλθε από το Ευρωπαϊκό Τηλεπικοινωνιακό Συμβούλιο το 1982, ενώ η αρχική του ονομασία ήταν Group Special Mobile. Το GSM έγινε εμπορικά διαθέσιμο στην Ευρώπη το 1991. Δυο χρόνια μετά την έκδοσή του, υπήρχαν 36 δίκτυα GSM σε πάνω από 22 χώρες της Ευρώπης. Το σύστημα αυτό υιοθετήθηκε και από άλλες χώρες στην Μέση Ανατολή, την Νότια Αφρική και την Αυστραλία.

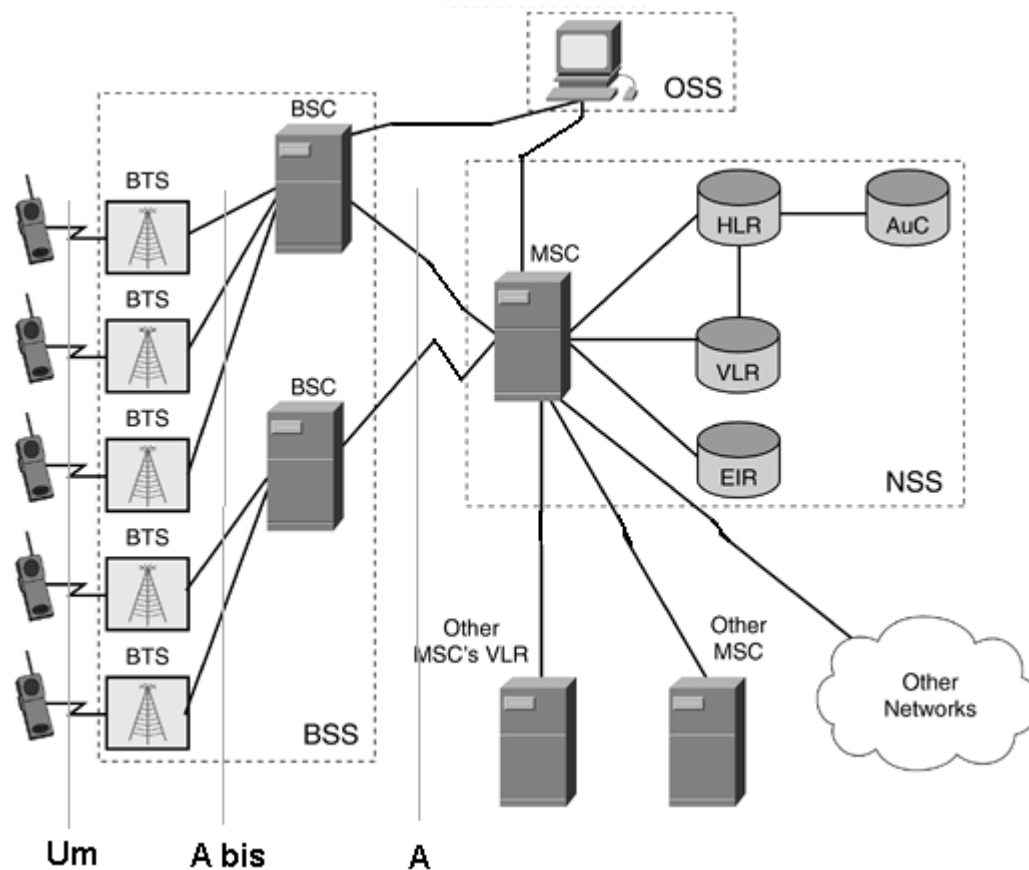
1.7 Δομή του GSM

Το GSM είναι ένα κυψελοειδές δίκτυο. Χρησιμοποιεί κυψέλες και σταθμούς βάσης για την τηλεπικοινωνία. Οι 5 πιο συνηθισμένες συχνότητες που χρησιμοποιεί είναι οι GSM-900, GSM-1800, GSM-850, GSM-1900 και GSM-400.

Ένα δίκτυο GSM αποτελείται από:

- Κινητό Σταθμό (Mobile Station – MS)
- Υποσύστημα Σταθμού Βάσης (Base Station Subsystem – BSS)
- Υποσύστημα Δικτύου και Μεταγωγής (Network and Switching Subsystem – NSS)
- Υποσύστημα Λειτουργίας και Υποστήριξης (Operation and Support Subsystem – OSS)

Στην εικόνα 1.8 φαίνεται διαγραμματικά η αρχιτεκτονική ενός δικτύου GSM.



Εικόνα 1.8: Η αρχιτεκτονική ενός δικτύου GSM.

Πηγή: <http://keepontouch-en.blogspot.gr>, Άρθρο *GSM Architecture and Definitions*

1.7.1 Κινητός Σταθμός

Ο Κινητός Σταθμός είναι ο φυσικός εξοπλισμός που χρησιμοποιείται για την επικοινωνία. Αυτός είναι η συσκευή του κινητού τηλεφώνου, η οποία συμπεριλαμβάνει το hardware της συσκευής που απαιτείται για την επικοινωνία και την κάρτα SIM, η οποία περιέχει τα στοιχεία του χρήστη. Επιπλέον, μια κινητή συσκευή έχει έναν αριθμό που λειτουργεί ως ταυτότητα και είναι μοναδικός στην κάθε μια, γνωστό και ως International Mobile Equipment Identity (IMEI).

1.7.2 Υποσύστημα Σταθμού Βάσης

Το κομμάτι αυτό είναι υπεύθυνο για την επικοινωνία με τους MSs του δικτύου. Συμπεριλαμβάνει τον απαραίτητο εξοπλισμό για τη μετάδοση και λήψη του σήματος, και για την διαχείριση αυτού. Αποτελείται από τον Σταθμό Βάσης Πομποδέκτη και τον Διαχειριστή Σταθμού Βάσης.

Ο Σταθμός Βάσης Πομποδέκτη (Base Transceiver Station – BTS) περιλαμβάνει τους ραδιοφωνικούς πομπούς, τις κεραίες τους, και γενικώς τον ευρύτερο εξοπλισμό με τον οποίο επιτυγχάνεται η επικοινωνία με τις κινητές συσκευές. Επιπρόσθετα, αναλαμβάνει την κρυπτογράφηση και την αποκρυπτογράφηση των σημάτων που αποστέλλονται για ασφάλεια. Είναι καθοριστικός για κάθε κυψέλη καθώς λειτουργεί σαν κόμβος. Η επικοινωνία του σταθμού με τα κινητά γίνεται μέσω της διεπαφής Um, με τα πρωτόκολλα της.

Ο Διαχειριστής Σταθμού Βάσης (Base Station Controller – BSC) ελέγχει έναν ή περισσότερους BTSs. Είναι υπεύθυνος για την διαχείριση των καναλιών (διανομή και απελευθέρωση), καθώς και για τη διαχείριση των μεταβιβάσεων από κυψέλη σε κυψέλη (handover) μεταξύ των BTSs που ελέγχει. Επικοινωνεί με τις BTSs μέσω της διεπαφής Abis.

1.7.3 Υποσύστημα Δικτύου και Μεταγωγής

Σε αυτό το υποσύστημα υπάρχουν πολλά στοιχεία. Αποτελείται από βάσεις δεδομένων και συστήματα για διαβιβάσεις των κλήσεων και των μηνυμάτων στους κατάλληλους σταθμούς βάσης. Χειρίζεται την επικοινωνία των συνδρομητών, παρέχει τον κύριο έλεγχο και τις διασυνδέσεις, και για αυτό το λόγο ονομάζεται συχνά κεντρικό δίκτυο. Αποτελείται από 7 στοιχεία.

Το κυριότερο ίσως στοιχείο μέσα στην αρχιτεκτονική του δικτύου GSM είναι το Κέντρο Υπηρεσιών Διαβίβασης Κινητών (Mobile Switching services Center – MSC). Λειτουργεί ως κόμβος παροχής υπηρεσιών, διαχειρίζεται τους διαθέσιμους πόρους και είναι υπεύθυνο για τη δρομολόγηση των κλήσεων εντός και εκτός του δικτύου του παρόχου, των SMS και παραπλήσιων υπηρεσιών.

Ο Καταχωρητής Τοποθεσίας (Home Location Register – HLR) είναι μια βάση δεδομένων η οποία περιέχει όλες τις πληροφορίες του χρήστη και της τρέχουσας τοποθεσίας του. Με αυτό τον τρόπο, το δίκτυο τον εντοπίζει γρηγορότερα και μεταφέρει την εκάστοτε κλήση στο πλησιέστερο σταθμό βάσης.

Ο Καταχωρητής Τοποθεσίας Επισκέπτη (Visitor Location Register – VLR) είναι μια προσωρινή βάση δεδομένων που αντλεί συγκεκριμένες πληροφορίες από το HLR. Οι πληροφορίες αυτές δίνουν κατ' επέκταση τη δυνατότητα σε ένα συνδρομητή να χρησιμοποιήσει συγκεκριμένες υπηρεσίες. Με αυτό τον τρόπο, μειώνει την υπερφόρτωση του HLR με αιτήσεις.

Ο Καταχωρητής Ταυτότητας Εξοπλισμού (Equipment Identity Register – EIR) είναι μια βάση δεδομένων που περιέχει πληροφορίες για την ταυτότητα των κινητών συσκευών του δικτύου του. Πραγματοποιεί έλεγχο ταυτότητας IMEI μιας συσκευής που εισέρχεται στο δίκτυο και αποφασίζει εάν θα επιτρέψει την πρόσβαση, εάν θα την απορρίψει ή αν θα την παρακολουθεί.

Το Κέντρο Ταυτοποίησης (Authentication Center – AuC) είναι μια προστατευόμενη βάση δεδομένων με σκοπό την ταυτοποίηση του χρήστη και την κρυπτογράφηση της επικοινωνίας του. Η ταυτοποίηση πραγματοποιείται κατά την προσπάθεια εισόδου του χρήστη στο δίκτυο με έναν έλεγχο του κλειδιού που βρίσκεται στην κάρτα SIM του εκάστοτε χρήστη. Εάν υπάρχει στη βάση δεδομένων, παρέχεται στο χρήστη ένα μοναδικό κλειδί κρυπτογράφησης για ασφάλεια επικοινωνίας.

Το Διαβιβαστικό Κέντρο Μεταγωγής Κινητών (Gateway Mobile Switching Center – GMSC) είναι ο διαβιβαστής των κλήσεων. Δρομολογούνται από αυτό χωρίς να είναι γνωστή η τοποθεσία του συνδρομητή, οπότε είναι αρμόδιο για τη συλλογή πληροφοριών τοποθεσίας μέσω του HLR και τη δρομολόγηση μιας κλήσης στο σωστό MSC.

Η Πύλη Γραπτών Μηνυμάτων (SMS Gateway – SMS-G) είναι ο όρος που χρησιμοποιείται για να περιγράψει 2 υπηρεσίες των SMS, μέσα στα πρότυπα του GSM. Πρόκειται για 2 διαβιβαστές σε διαφορετικές κατευθύνσεις. Το SMS-GMSC (Short Message Service Gateway Mobile Switching Center) που είναι υπεύθυνο για SMS που αποστέλλονται σε κινητές συσκευές, και το SMS-IWMSC (Short Message Service Inter Working Mobile Switching Center), που χρησιμοποιείται για SMS που έχουν δημιουργηθεί από μια κινητή συσκευή στο δίκτυο. Με λίγα λόγια, το SMS-GMSC είναι παρόμοιο στη λειτουργία του με το GMSC και το SMS-IWMSC παρέχει ένα σταθερό σημείο πρόσβασης στο Κέντρο Διαχείρισης SMS.

1.7.4 Υποσύστημα Λειτουργίας και Υποστήριξης

Το OSS είναι ένας λειτουργικός φορέας, μέσω του οποίου ο διαχειριστής του δικτύου ελέγχει και το παρακολουθεί το δίκτυο. Ο σκοπός του είναι να προσφέρει αποδοτική υποστήριξη για τις απαραίτητες δραστηριότητες συντήρησης που απαιτούνται σε ένα δίκτυο GSM.

1.8 Ορισμός ενός αισθητήρα

Ένας αισθητήρας είναι ένα μηχανήμα ή συσκευή, το οποίο έχει την δυνατότητα να συλλέγει πληροφορίες από ένα φυσικό δεδομένο όπως κίνηση, θερμοκρασία, ήχο, εικόνα, και να το μετατρέπει σε μορφές και σήματα που μπορούν να διαβαστούν από τον άνθρωπο ή από άλλα ηλεκτρονικά μέσα. Ένας αισθητήρας ανταποκρίνεται σε ένα μόνο στοιχείο, το οποίο δεν επηρεάζεται από τη μέτρηση.

Στη ζωή ενός ανθρώπου υπάρχουν αισθητήρες σε πολλά αντικείμενα και μηχανήματα καθημερινής χρήσης. Για παράδειγμα σε ένα καινούριο αυτοκίνητο, υπάρχουν διάφοροι αισθητήρες όπως θερμομέτρο, αισθητήρας παρκαρίσματος, αισθητήρες ταχύτητας και επιτάχυνσης, σε μια κάμερα υπάρχουν αισθητήρες για καταγραφή εικόνας και ήχου, ενώ μεγάλη πληθώρα αισθητήρων υπάρχει πλέον και στα κινητά τηλέφωνα.

1.9 Αισθητήρες σε κινητά τηλέφωνα

Οι κινητές συσκευές είναι εξοπλισμένες με αισθητήρες με αρχικό σκοπό την εξυπηρέτηση των αναγκών του χρήστη για ομιλία. Πέραν της ομιλίας όμως, ένα smartphone με τη χρήση των κατάλληλων αισθητήρων λειτουργεί και ως φωτογραφική μηχανή, ως συσκευή πλοήγησης με τη χρήση του GPS, ως καταγραφέας φωνής, ενώ με εφαρμογές τα δεδομένα που εισάγονται στη συσκευή μέσω των αισθητήρων μπορούν να χρησιμοποιηθούν με διάφορους τρόπους προς όφελος του χρήστη. Οι αισθητήρες ανήκουν στο τμήμα του hardware μιας συσκευής. Οι παρακάτω αισθητήρες είναι αυτοί που συναντώνται πιο συχνά στα smartphones.

1.9.1 Οθόνη Αφής

Η οθόνη αφής (touch screen) είναι ένας συνδυασμός δυο διαφορετικών τεχνολογιών. Αυτή της οθόνης, μιας συσκευής εξόδου, όπου απεικονίζονται ηλεκτρονικά εικόνες και βίντεο σε μια επίπεδη επιφάνεια και αυτή της διαδραστικότητας μέσω της αφής, μιας συσκευής εισόδου, όπου ο χρήστης μπορεί να χειριστεί μια συσκευή με απλές κινήσεις του χεριού επάνω στην επιφάνεια αφής, η οποία τις αναγνωρίζει και συλλέγει πληροφορίες. Παλαιότερα υπήρχαν οθόνες αφής που λειτουργούσαν με πίεση, οπότε ήταν ευκολότερη η χρήση τους με γραφίδα. Πλέον δεν χρειάζεται πίεση για να ανιχνευτεί μια κίνηση. Η οθόνη

αποτελείται από πολλά στρώματα ενωμένα, τα οποία περιλαμβάνουν μια οθόνη υγρών κρυστάλλων (Liquid Crystal Display – LCD), προστατευτικά φίλτρα και γυαλί, και αισθητήρες κίνησης.

1.9.2 Μικρόφωνο

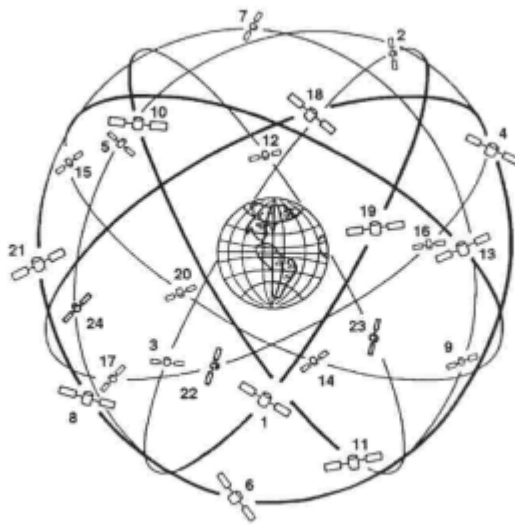
Το μικρόφωνο είναι μια συσκευή που δέχεται ηχητικά κύματα και τα μετατρέπει σε ηλεκτρικά σήματα για τη δυνατότητα καταγραφής και μετάδοσης. Είναι απαραίτητος αισθητήρας σε ένα κινητό τηλέφωνο για την τηλεφωνική επικοινωνία του χρήστη, αλλά πλέον υπάρχουν εφαρμογές που εκμεταλλεύονται το μικρόφωνο για απλά πράγματα όπως καταγραφή ήχου, μέχρι πιο περίπλοκα όπως αναγνώριση μουσικών κομματιών με την ηχογράφηση ενός μικρού μέρους τους.

1.9.3 Ψηφιακή Φωτογραφική Μηχανή (Κάμερα)

Η ψηφιακή φωτογραφική μηχανή χρησιμοποιείται περισσότερο με την αγγλική της ονομασία, κάμερα (camera). Πρόκειται για μια συσκευή καταγραφής εικόνας (ή και βίντεο) ηλεκτρονικά. Αποτελείται από έναν φακό που συγκεντρώνει φως, το οποίο με τη σειρά του μετατρέπεται σε ηλεκτρικό σήμα από τον αισθητήρα για την αποθήκευση ή τη χρήση του από τη συσκευή. Υπάρχουν δυο είδη αισθητήρα, ο Complementary Metal Oxide Semiconductor (CMOS) και ο Charge Couple Device (CCD). Συνήθως η κάμερα συνοδεύεται από φλας για υποβοηθούμενο φωτισμό σε καταστάσεις έλλειψης ή μη αρκετής ύπαρξης αυτού. Σε συνδιασμό με άλλους αισθητήρες, η κάμερα αποκτά περισσότερες δυνατότητες. Ο χρήστης μπορεί να εστιάσει σε διαφορετικά αντικείμενα αγγίζοντάς τα στην οθόνη πρώτου γίνι η λήψη της εικόνας, ενώ μπορούν να αποθηκευτούν φωτογραφίες με ενσωματωμένο το στίγμα της τοποθεσίας τους μέσω του GPS.

1.9.4 GPS

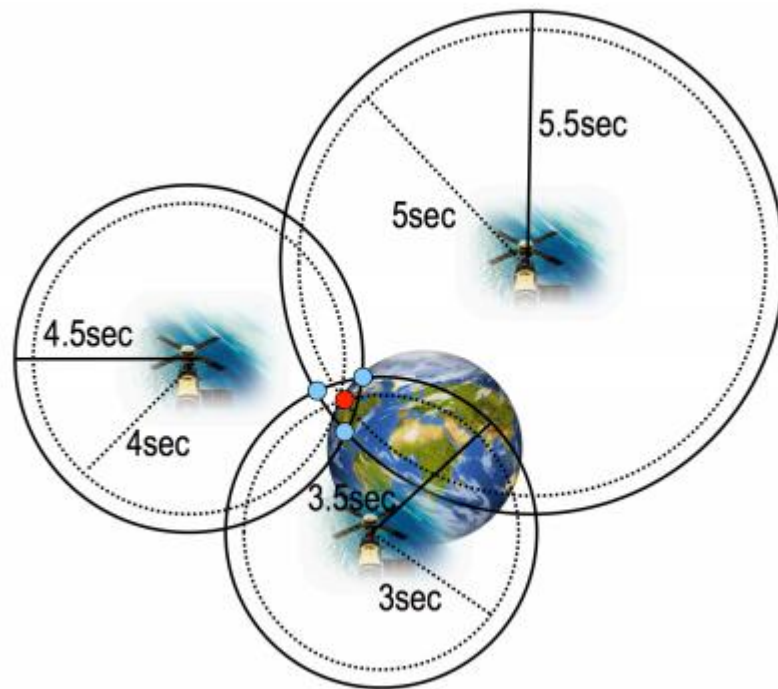
Στα ελληνικά ονομάζεται Παγκόσμιο Σύστημα Θεσιθεσίας, αλλά ομοίως με την κάμερα χρησιμοποιείται η αγγλική ονομασία του, Global Positioning System ή GPS. Είναι ένα σύστημα εντοπισμού θέσης σε συνεργασία με 27 δορυφόρους που περιφέρονται γύρω από τη Γη (εικόνα 1.9).



Εικόνα 1.9: Η τοποθέτηση των δορυφόρων γύρω από τη Γη.

Πηγή: <http://www.globalpositioningsystem.de>, Άρθρο *Global Positioning System (GPS)*

Ο αριθμός και η τοποθέτηση των δορυφόρων είναι τέτοια ώστε να είναι πάντα ορατοί 4 δορυφόροι σε οποιοδήποτε σημείο της Γης βρίσκεται κάποιος. Για τον ακριβή προσδιορισμό της τοποθεσίας, χρειάζονται τουλάχιστον 3 δορυφόροι, εάν ο χρήστης επιθυμεί και ακριβές υψόμετρο τότε χρειάζονται 4. Όταν ο χρήστης ενεργοποιεί το GPS στο κινητό του, αποστέλλεται ένα σήμα από τη συσκευή το οποίο λαμβάνεται από τους ορατούς δορυφόρους. Έπειτα, ο κάθε δορυφόρος υπολογίζει πόση ώρα χρειάστηκε για να λάβει το σήμα και υπολογίζει σε ποιο σημείο της εμβέλειάς του βρίσκεται ο χρήστης. Με την συνεργασία των στοιχείων αυτών μεταξύ των δορυφόρων υπολογίζεται η ακριβής τοποθεσία του χρήστη (εικόνα 1.10) και επιστρέφεται στο κινητό του σαν στίγμα πάνω σε χάρτη ή σε αριθμούς γεωγραφικού μήκους και πλάτους.



Εικόνα 1.10: Προσδιορισμός ακριβούς τοποθεσίας με τη συνεργασία 3 δορυφόρων.

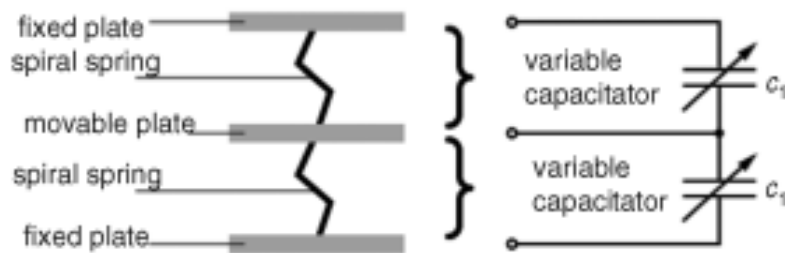
Πηγή: <http://www.uni-weimar.de>, Άρθρο *Smartphone Hardware Sensors*

Το GPS δεν χρειάζεται απαραίτητα σύνδεση στο internet μέσω WiFi ή 3G για να δουλέψει, παρόλα αυτά εάν ο χρήστης επιθυμεί να το χρησιμοποιήσει για πλοήγηση με τη χρήση χαρτών όπως του google maps, τότε χρειάζεται σύνδεση για να φορτώσουν οι χάρτες. Σε πολλά smartphones υπάρχει το Assisted GPS (A-GPS ή aGPS), το οποίο είναι ένα σύστημα υποβοηθούμενου GPS που αξιοποιεί τα δεδομένα του δικτύου μέσω των κεραιών για πιο γρήγορο υπολογισμό της θέσης, με την ίδια μέθοδο που χρησιμοποιούν οι δορυφόροι.

1.9.5 Επιταχυνσιόμετρο

Το επιταχυνσιόμετρο (accelerometer) είναι μια συσκευή μέτρησης δυνάμεων επιτάχυνσης. Η επιτάχυνση μετριέται σε 3 άξονες, μήκος, πλάτος και ύψος. Στα smartphones χρησιμοποιούνται χωρητικά επιταχυνσιόμετρα. Αποτελούνται από τρία φύλλα πυριτίου, τοποθετημένα παράλληλα και συνδεδεμένα με σπειροειδή ελατήρια (σχήμα 1.4), σχηματίζοντας έτσι μια σύνδεση πυκνωτών. Τα δυο εξωτερικά φύλλα είναι σταθερά, ενώ το μεσαίο μπορεί να μετακινείται, έχοντας έτσι το ρόλο της μάζας. Όταν το μεσαίο φύλλο μετακινείται, η χωρητικότητα των

πυκνωτών μεταβάλλεται και η μέτρηση αυτής μετατρέπεται σε μονάδα επιτάχυνσης.



Σχήμα 1.4: Ένα χωρητικό επιταχυνσιόμετρο.

Πηγή: <http://physicsandroid.wordpress.com>, Άρθρο *Αισθητήρας επιτάχυνσης*

Για την μέτρηση της επιτάχυνσης σε τρεις άξονες, τοποθετούνται στη συσκευή 3 τέτοιοι αισθητήρες, κάθετα μεταξύ τους. Το επιταχυνσιόμετρο χρησιμοποιείται για διάφορες εφαρμογές και λειτουργίες του τηλεφώνου. Η πιο συνηθισμένη λειτουργία του είναι η αυτόματη περιστροφή της οθόνης ανάλογα με την κλίση της συσκευής. Συνεισφέρει σε πολλές εφαρμογές, όπως αυτές που συμπεριλαμβάνουν βηματομετρητές και εφαρμογές παιχνιδιών καθώς παρέχει στο χρήστη μια πιο διαδραστική εμπειρία.

1.9.6 Γυροσκόπιο

Το γυροσκόπιο (gyroscope) είναι μια συσκευή που χρησιμοποιεί την βαρύτητα της Γης για να προσδιορίσει τον προσανατολισμό. Αποτελείται από έναν ελεύθερα κινούμενο δίσκο, τοποθετημένο επάνω σε έναν κινούμενο άξονα στο κέντρο της μεγαλύτερου τροχού. Στα smartphone έχει διαφορετικό σχεδιασμό καθώς το μέγεθος του αισθητήρα πρέπει να είναι όσο το δυνατόν μικρότερο. Χρησιμοποιεί ένα μικροηλεκτρονικομηχανικό σύστημα (microelectromechanical system – MEMS) για την ενσωμάτωση ηλεκτρονικών και μηχανικών κομματιών σε πολύ μικρή κλίμακα. Μια συσκευή MEMS αποτελείται από ειδικά προσαρμοσμένα κυκλώματα και έναν αισθητήρα σιλικόνης. Τα smartphones χρησιμοποιούν το γυροσκόπιο για να προσδιορίσουν τον προσανατολισμό της συσκευής. Πολύ συχνά συνεργάζεται με το επιταχυνσιόμετρο για μεγαλύτερη ακρίβεια.

1.9.7 Αισθητήρας φωτός

Ο αισθητήρας φωτός χρησιμοποιείται στα smartphones για προσαρμοσμένη ρύθμιση φωτεινότητας της οθόνης. Αποτελείται από διάφορους τύπους φωτοκυττάρων, ευαίσθητους στο φως. Είναι τοποθετημένος επάνω από την οθόνη του κινητού και παρέχει στο χρήστη τη δυνατότητα αυτόματης προσαρμογής της φωτεινότητας της οθόνης, ανάλογα με το περιβάλλον στο οποίο βρίσκεται για ευκολία στην προβολή και εξοικονόμηση ενέργειας.

1.9.8 Εγγύτητα

Ο αισθητήρας εγγύτητας έχει τη δυνατότητα να υπολογίζει την απόσταση της συσκευής από ένα άλλο σώμα. Στα smartphones χρησιμοποιείται για λόγους πρακτικότητας και εξοικονόμησης ενέργειας. Είναι τοποθετημένο επάνω από την οθόνη, όπως ο αισθητήρας φωτός. Χρησιμοποιεί αισθητήρες υπέρυθρων για τον υπολογισμό της απόστασης της συσκευής με κάποιο ξένο αντικείμενο. Όταν ο χρήστης τοποθετεί τη συσκευή στο αυτί του, ο αισθητήρας δίνει σήμα για την απενεργοποίηση της οθόνης ώστε να μην πατηθεί κάποιο κουμπί κατά λάθος αλλά και για εξοικονόμηση ενέργειας. Όταν η συσκευή απομακρυνθεί από το αυτί του χρήστη, η οθόνη επανέρχεται σε λειτουργία.

1.10 Tablets

Το tablet είναι ένας μικρός φορητός ηλεκτρονικός υπολογιστής, μικρότερο σε μέγεθος από ένα laptop και μεγαλύτερο από ένα smartphone (εικόνα 1.11). Έχει αισθητήρες όπως κάμερα, οθόνη αφής, μικρόφωνο, GPS, το πληκτρολόγιο είναι εικονικό και μπορεί να έχει μερικά κουμπιά για τη ρύθμιση της έντασης του ήχου και για κλείδωμα της συσκευής. Το λειτουργικό του σύστημα ποικίλει μεταξύ διάφορων εκδόσεων Android, IOS και Windows, με τις 2 πρώτες να συναντώνται πιο συχνά. Συνδέονται στο internet μέσω WiFi ή 3G.

Η κύρια διαφορά με τα smartphone είναι η μεγαλύτερη οθόνη στα tablet, της οποίας το μέγεθος κυμαίνεται από 7-10,5 ίντσες. Τα tablet έκαναν την πρώτη τους εμφάνιση το 2001 με το Microsoft Tablet PC, ενώ έγιναν δημοφιλή μετά το 2010 όπου κυκλοφόρησε το iPad της Apple.



Εικόνα 1.11: Εμφάνιση ενός tablet με λειτουργικό σύστημα Android.

Πηγή: <http://www.t3.com.au>, Άρθρο *Samsung Galaxy Tab 7.0 Plus to launch in US November 13th*

ΚΕΦΑΛΑΙΟ 2

Λειτουργικά Συστήματα Κινητών

2.1 Εισαγωγή

Μια συσκευή κινητού τηλεφώνου δεν μπορεί να λειτουργήσει χωρίς κάποιο λειτουργικό σύστημα. Αποτελεί ζωτικό κομμάτι που συνδυάζει το υλικό μιας συσκευής, αξιοποιώντας και συνδυάζοντας τις δυνατότητές του. Στην υψηλότερη θέση του ανταγωνισμού εμφανίζεται το Android, λόγω του ανοιχτού κώδικα λογισμικού του, το οποίο χρησιμοποιήθηκε και για τη δημιουργία της εφαρμογής.

2.2 Ορισμός Λειτουργικού Συστήματος

Ένα Λειτουργικό Σύστημα (Operating System – OS), ή OS όπως αναφέρεται συχνότερα, είναι το λογισμικό που είναι υπεύθυνο για την διαχείριση των πόρων ενός ηλεκτρονικού υπολογιστή. Οι πόροι αυτοί περιέχουν το υλικό (hardware) και το λογισμικό (software). Είναι ένα σημαντικό κομμάτι του υπολογιστή, καθώς οι εφαρμογές συνήθως απαιτούν ένα λειτουργικό σύστημα για να λειτουργήσουν.

Ένα λειτουργικό σύστημα μπορεί να υπάρχει σε οποιαδήποτε συσκευή που περιέχει κάποιου είδους υπολογιστή, από κινητά τηλέφωνα μέχρι παιχνιδομηχανές. Στην κατηγορία των κινητών τηλεφώνων, τα δημοφιλέστερα λειτουργικά συστήματα είναι:

- Android OS
- iOS
- Blackberry OS
- Symbian
- Windows Mobile

2.3 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για κινητές συσκευές, βασισμένο στο πρότυπο των Linux. Οι πιο διαδεδομένες συσκευές που συναντάται είναι τα smartphones και τα tablets. Έχει το δικό του Kit Ανάπτυξης Λογισμικού

(Software Development Kit – SDK) το οποίο παρέχεται δωρεάν και η σύνθεση του κώδικα γίνεται με τη χρήση της γλώσσας προγραμματισμού Java.

2.4 Ιστορική Αναδρομή

Το Android αναπτύχθηκε αρχικά από την Android Inc., η οποία εξαγοράστηκε από την Google το 2005. Έκανε την εμφάνισή του μέσα από μια συνεργασία με μέλη της Open Handset Alliance, μιας κοινοπραξίας 84 εταιρειών τεχνολογίας και κινητών τηλεφώνων, όπου τελειοποιήθηκε και κυκλοφόρησε τελικά το Νοέμβριο του 2007. Αξίζει να αναφερθούν ενδεικτικά μερικά μεγάλα ονόματα της κοινοπραξίας αυτής, γιατί με το κύρος του ονόματός τους έδωσαν προοπτική στο έργο τους.

Δέκα μεγάλα ονόματα της κοινοπραξίας:

- § T-Mobile
- § Ebay
- § Google
- § Intel
- § HTC
- § LG
- § Motorola
- § Nvidia
- § Samsung
- § Synaptics

Ο κοινός στόχος των μελών της κοινοπραξίας ήταν η παραγωγή μιας αναπτυσσόμενης καινοτομίας για κινητές συσκευές με σαφώς καλύτερη εμπειρία χρήστη σε σύγκριση με τα υπάρχοντα δεδομένα της εποχής εκείνης.

Το κύριο πρόβλημα ήταν η δυσκολία και η έλλειψη συλλογικής εργασίας με κοινό απώτερο σκοπό. Αυτό το πρόβλημα ήρθε να λύσει το Android, έχοντας μια από τις πιο προοδευτικές άδειες, φιλικές προς το χρήστη με λογισμικό ανοιχτού κώδικα, οι οποίες δίνουν μια σημαντική ελευθερία κινήσεων και ευελιξία στους παρόχους και στους κατασκευαστές κινητής τηλεφωνίας για την δημιουργία νέων προϊόντων.

Το λογότυπο του Android είναι ένα ρομπότ σε ανοιχτό πράσινο χρώμα, σχεδιασμένο από τη γραφίστρια Irina Blok (εικόνα 2.1).



Εικόνα 2.1: Το λογότυπο της πλατφόρμας Android.

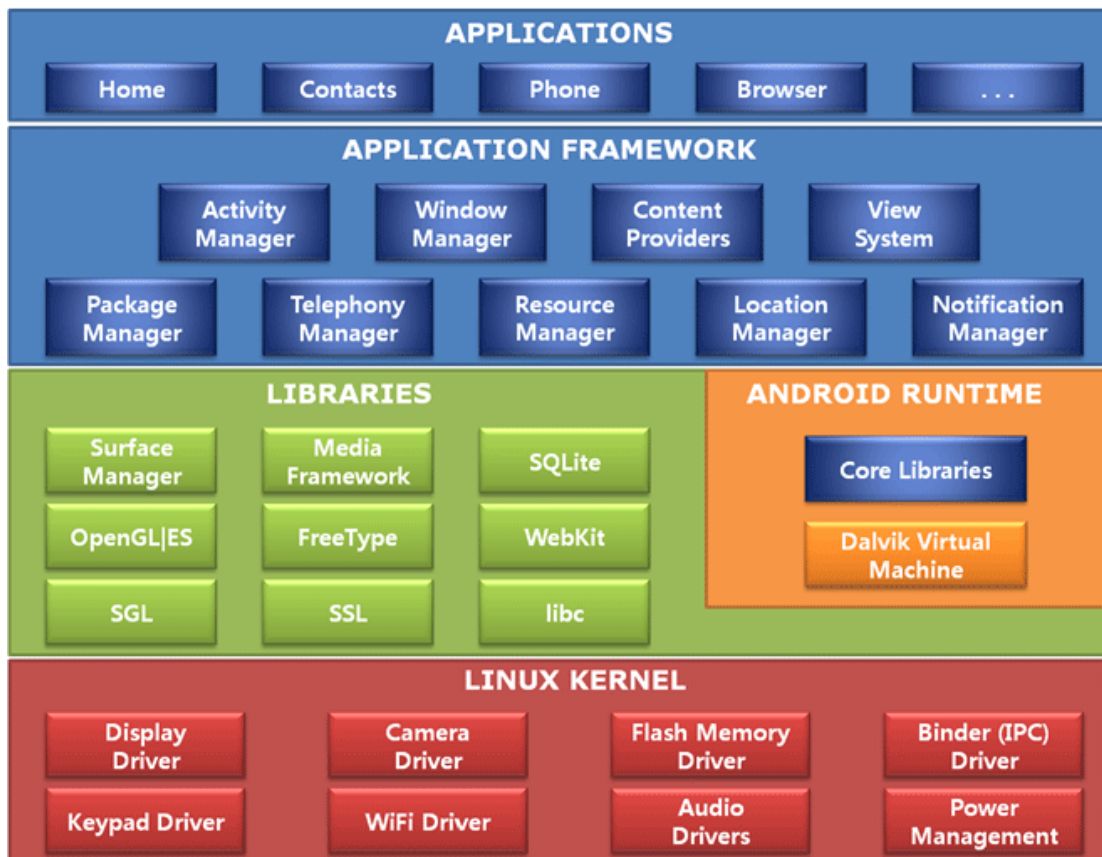
Πηγή: <http://famouslogos.net>, Άρθρο *Android Logo*

2.5 Ορισμός του API

Το Περιβάλλον Προγραμματισμού Εφαρμογών, ευρέως γνωστό ως API (Application Programming Interface) είναι ένα σύνολο από λειτουργίες, ρουτίνες (routines), πρωτόκολλα (protocols) και εργαλεία για τη δημιουργία εφαρμογών λογισμικού. Η δουλειά του API είναι να καθορίζει το πως θα αλληλεπιδρούν και θα χρησιμοποιούνται τα εξαρτήματα του λογισμικού κατά των προγραμματισμό Γραφικού Περιβάλλοντος Χρήστη (Graphical User Interface - GUI). Το επίπεδο του API ορίζει το εύρος των επιλογών, την ευκολία δημιουργίας και τις δυνατότητες και παρέχει στην ουσία τα δομικά στοιχεία στον προγραμματιστή, ο οποίος καλείται να τα χρησιμοποιήσει όπως αυτός επιθυμεί για να “χτίσει” την εφαρμογή του. Όσο μεγαλύτερο το επίπεδο του API, τόσο περισσότερα και καλύτερα πράγματα μπορεί να προσθέσει κάποιος στην εφαρμογή του.

2.6 Αρχιτεκτονική του Android

Το Android, για να γίνει κατανοητή η αρχιτεκτονική του, μπορεί να θεωρηθεί σαν μια στοίβα λογισμικού που περιλαμβάνει διάφορα επίπεδα. Το κάθε επίπεδο ακολουθεί μια καθορισμένη συμπεριφορά και παρέχει ειδικές υπηρεσίες στο επίπεδο που βρίσκεται πάνω από αυτό. Στην εικόνα 2.2 απεικονίζονται τα επίπεδα αυτά, καθώς και τα στοιχεία που περιλαμβάνουν.



Εικόνα 2.2: Η στοίβα του Android με τα 5 επίπεδα και τα στοιχεία αυτών.

Πηγή: <http://android.blog.ir>, Άρθρο *Android Architecture*

Στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας Linux (kernel), ο οποίος χρησιμοποιείται από το Android. Επάνω από τον Linux kernel βρίσκονται οι βιβλιοθήκες (Libraries) και ο χρόνος εκτέλεσης (Android Runtime). Πάνω απ' αυτά βρίσκεται το πλαίσιο εφαρμογής (Application Framework), το οποίο επιτρέπει στο λειτουργικό σύστημα να αλληλεπιδρά με τον πυρήνα και τις βιβλιοθήκες. Στο υψηλότερο επίπεδο βρίσκονται οι εφαρμογές (Applications).

2.6.1 Linux Kernel

Πρόκειται για τον πυρήνα πάνω στον οποίο βασίζεται το λειτουργικό σύστημα. Βρίσκεται στο χαμηλότερο επίπεδο για να παρέχει μια γερή βάση. Δεν είναι το παραδοσιακό Linux που συναντάται σε άλλες συσκευές, το Android έχει τροποποιήσει τον κώδικα του πυρήνα του Linux, έτσι ώστε να τρέχει σε ένα συγκεκριμένο περιβάλλον. Αυτό σημαίνει ότι δεν συμπεριλαμβάνονται όλα τα χαρακτηριστικά του συνηθισμένου πακέτου Linux. Οι αλλαγές που έχουν γίνει οφείλονται στους προγραμματιστές του Android, και επειδή έχει τροποποιηθεί θετικά τόσο πολύ ο αρχικός κώδικας, η κοινότητα του Linux αποφάσισε να συμπεριλάβει αυτές τις αλλαγές στην επόμενη έκδοση του.

Το Android βασίζεται στον πυρήνα για όλα τα στοιχειώδη όπως πρόσβαση στα προγράμματα οδήγησης (drivers) και υλικού της συσκευής (hardware), στην παροχή προστασίας και στη διαχείριση υπηρεσιών και μνήμης. Χρησιμοποιώντας το Linux ως συνδετικό κρίκο μεταξύ του υλικού κομματιού μιας συσκευής και του λειτουργικού συστήματος, ανοίγεται ο δρόμος στο Android για δυνατότητα χρήσης του σε μια πληθώρα συσκευών.

2.6.2 Libraries

Το Android περιλαμβάνει ένα σετ βιβλιοθηκών, γραμμένες σε κώδικα C και C++, οι οποίες χρησιμοποιούνται από διάφορα στοιχεία (components) του συστήματος. Οι προγραμματιστές χρησιμοποιούν αυτές τις βιβλιοθήκες μέσω του πλαισίου εφαρμογών του Android. Το επίπεδο αυτό αναφέρεται και ως Native Libraries (εγγενείς βιβλιοθήκες). Ο κώδικας είναι γραμμένος σε C/C++ και βελτιστοποιημένος για το hardware, σε αντίθεση με τις εφαρμογές του Android και το framework που είναι γραμμένα σε Java. Η επικοινωνία μεταξύ των εφαρμογών και των εγγενών δυνατοτήτων γίνονται μέσω του Java Native Interface (JNI), ενός πλαισίου προγραμματισμού που επιτρέπει σε ένα σύστημα να λειτουργεί με Java η οποία μπορεί να καλεί ή να καλείται από εγγενείς εφαρμογές και βιβλιοθήκες γραμμένες σε άλλες γλώσσες.

Επεξήγηση βιβλιοθηκών:

- Surface Manager: διαχειρίζεται την πρόσβαση στο υποσύστημα οθόνης και συνθέτει επίπεδα γραφικών 2D και 3D από πολλαπλές εφαρμογές

- **Media framework:** υποστηρίζει την εγγραφή και αναπαραγωγή πολλών format εικόνας και ήχου (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG). Είναι βασισμένο στο OpenCore του PacketVideo
- **SQLite:** παρέχει σχεσιακές βάσεις δεδομένων που μπορούν να χρησιμοποιηθούν από τις εφαρμογές και τα συστήματα
- **OpenGL|ES:** είναι μια βιβλιοθήκη για γραφικά 3D. Έχει μια εφαρμογή λογισμικού που επιταχύνει το hardware της συσκευής εάν είναι συμβατή με τα 3D γραφικά.
- **FreeType:** μηχανή παροχής γραμματοσειρών
- **WebKit:** browser ανοιχτού κώδικα
- **SGL:** η μηχανή γραφικών 2D
- **SSL :** παρέχει υποστήριξη για τυπικές κρυπτογραφημένες λειτουργίες
- **Libc:** η πρότυπη βιβλιοθήκη για τη γλώσσα προγραμματισμού C

2.6.3 Android Runtime

Σε αυτό το επίπεδο βρίσκονται δυο διαφορετικά components, μια εικονική μηχανή (Dalvik VM) και οι βιβλιοθήκες πυρήνα (Core Libraries).

Η Dalvik είναι ουσιαστικά ένα λογισμικό ανοιχτού κώδικα, και είναι υπεύθυνη για τη λειτουργία των εφαρμογών. Οι εφαρμογές του Android είναι γραμμένες σε Java και μετά το compile γίνονται αρχεία Java class. Επειδή το Android δεν τρέχει αυτές τις κλάσεις αυτούσιες, τα αρχεία περνάνε από δεύτερο compile και αλλάζει το format τους σε dex, το οποίο τρέχει μέσω της Dalvik. Η μορφή των αρχείων είναι πιο συμπαγής και έτσι εξοικονομείται αποθηκευτικός χώρος.

Οι Core Libraries παρέχουν το μεγαλύτερο μέρος της λειτουργικότητας των API.

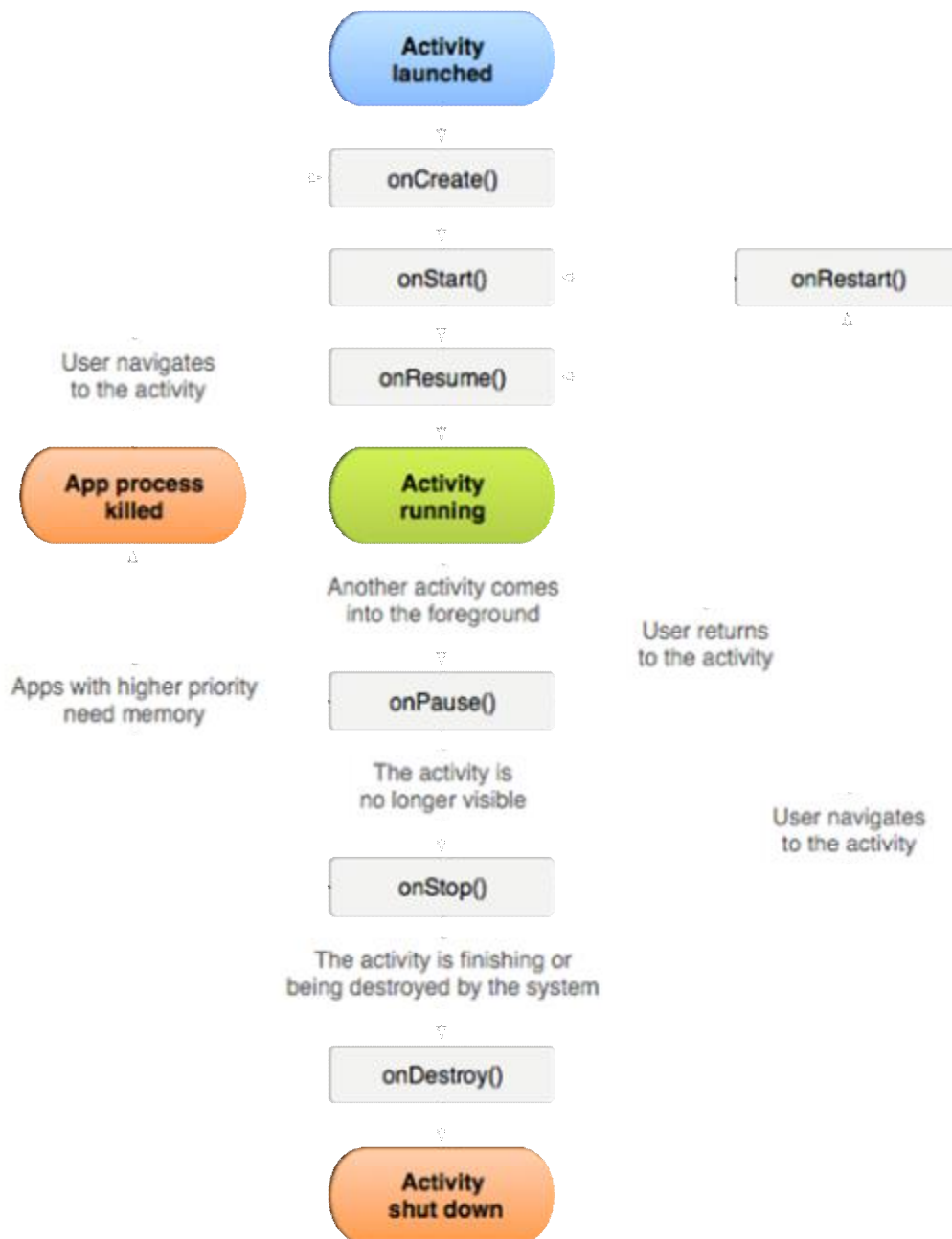
Οι βιβλιοθήκες αυτές δεν εξαρτώνται από το Android, αλλά είναι ένα σημαντικό κομμάτι στα θεμέλια του. Το μπλε χρώμα στην εικόνα 2.2 υποδηλώνει ότι είναι γραμμένες σε Java.

2.6.4 Application Framework

Το επίπεδο αυτό είναι γραμμένο σε Java. Εδώ παρέχεται ένα σετ από κλάσεις υψηλότερου επιπέδου, που προσφέρονται στους προγραμματιστές μέσω των API, των πιο σημαντικών δομικών στοιχείων για την δημιουργία εφαρμογών. Είναι ένα

εργαλείο που χρησιμοποιούν όλες οι εφαρμογές. Τα πιο σημαντικά στοιχεία σε αυτό το επίπεδο είναι:

- Activity Manager: διαχειριστής κύκλου ζωής των δραστηριοτήτων (σχήμα 2.1) και υπεύθυνο για την ομαλή λειτουργία εφαρμογών που τρέχουν ταυτόχρονα
- Package Manager: διαχειριστής των εγκατεστημένων εφαρμογών στη συσκευή και των δυνατοτήτων τους
- Window Manager: διαχειριστής παραθύρων
- Telephony Manager: διαχειρίζεται τα API που σχετίζονται με την τηλεφωνική επικοινωνία της συσκευής
- Content Providers: διαχειριστές του μοιρασμού δεδομένων μεταξύ των εφαρμογών
- Resource Manager: διαχειριστής των διαφόρων αρχείων που χρησιμοποιούνται στις εφαρμογές
- View System: περιλαμβάνει όλα τα δομικά στοιχεία του περιβάλλοντος χρήστη, όπως κουμπιά και λίστες
- Location Manager: διαχείριση τοποθεσίας με τη χρήση GPS ή μέσω του δικτύου κινητής τηλεφωνίας
- Notification Manager: διαχειριστής ειδοποιήσεων



Σχήμα 2.1: Ο κύκλος ζωής μιας δραστηριότητας Android.

Πηγή: <http://developer.android.com>, Άρθρο *Activity*

2.6.5 Applications

Στο υψηλότερο επίπεδο βρίσκονται οι εφαρμογές. Το Android κυκλοφορεί πάντα με προεγκατεστημένες εφαρμογές όπως τον browser, τα SMS, το ημερολόγιο, την υπηρεσία e-mail, τους χάρτες της Google, τις επαφές του τηλεφώνου, και διάφορα άλλα. Ο χρήστης έχει τη δυνατότητα να κατεβάσει περισσότερες εφαρμογές της αρεσκείας του μέσω του Google Play, επωφελούμενος της μεγάλης γκάμας

εφαρμογών όλων των ειδών που κυκλοφορούν. Η διαμόρφωση μιας συσκευής γίνεται μέσω των εφαρμογών διότι αυτές δίνουν μια πιο προσωπική νότα που αντιπροσωπεύει τις ανάγκες και τις προτιμήσεις του καθενός. Το Android συμπεριφέρεται σε όλες τις εφαρμογές το ίδιο, είτε έχουν δημιουργηθεί από την ίδια την εταιρεία, είτε από ανεξάρτητους προγραμματιστές. Όλες οι εφαρμογές είναι προγραμματισμένες σε Java.

2.7 Εκδόσεις Android

Η πρώτη έκδοση του Android (Android beta) ήταν στις 5 Νοεμβρίου του 2007, με την πρώτη έκδοση του SDK να ακολουθεί μια εβδομάδα αργότερα. Η επόμενη έκδοση ήταν και η πρώτη εμπορικά διαθέσιμη, με την ονομασία Android 1.0, η οποία έκανε την εμφάνισή της τον Οκτώβριο του 2008 με το πρώτο smartphone που χρησιμοποιούσε το λειτουργικό σύστημα Android. Το κινητό αυτό ήταν το T-Mobile G1, γνωστό και ως HTC Dream, προϊόν της εταιρείας HTC (εικόνα 2.3).



Εικόνα 2.3: T-Mobile G1, το πρώτο κινητό με λογισμικό Android.

Πηγή: <http://www.chip.de>, Άρθρο *Die Models und die Freaks*

Στην έκδοση αυτή κάνει την εμφάνισή του η εφαρμογή Android Market, στην οποία μπορεί ο χρήστης να βρει και να κατεβάσει στο κινητό του άλλες εφαρμογές. Διαθέτει επίσης browser, κάμερα, συγχρονισμό με παροχές της Google όπως το Calendar (Ημερολόγιο), τις επαφές και το Gmail, καθώς και τις εφαρμογές Google Maps και Youtube.

Ακολουθεί η έκδοση Android 1.1, γνωστή και ως “Petit Four”, το Φεβρουάριο του 2009. Στην έκδοση αυτή δεν υπάρχει μεγάλη διαφορά από την πρώτη, έχουν διορθωθεί σφάλματα και παρατηρείται αλλαγή στο API από 1 σε 2.

Τον Απρίλιο του 2009 βγήκε η έκδοση Android 1.5 με την ονομασία Cupcake (εικόνα 2.4), η πρώτη επίσημη έκδοση με ονομασία γλυκού. Οι επόμενες εκδόσεις ακολουθούν αυτή την “παράδοση”, χρησιμοποιώντας το επόμενο γράμμα της αλφαβήτου κάθε φορά. Το Cupcake κάνει χρήση του Linux kernel 2.6.27.



Εικόνα 2.4: Το λογότυπο του Android 1.5 Cupcake.

Πηγή: <http://www.gsmdome.com>, Άρθρο *Android 1.5 Gets Announced, Detailed*

Soon

Το Cupcake έφερε μεγάλες αλλαγές στο περιβάλλον χρήστη (User Interface) και στη χρήση:

- Μαλακό πληκτρολόγιο αφής στην οθόνη
- Δυνατότητα εγγραφής και αναπαραγωγής βίντεο καθώς και μεταφόρτωσής τους στο Youtube ή φωτογραφιών στο Picasa
- Νέα εμφάνιση σε εικονίδια εφαρμογών και widget
- Γρηγορότερη εύρεση τοποθεσίας μέσω GPS
- Κινούμενες μεταβάσεις οθόνης
- Δυνατότητα σύζευξης με άλλες συσκευές μέσω Bluetooth με τη χρήση των προφίλ A2DP και AVCRP.

Η επόμενη έκδοση είναι το Android 1.6, με την ονομασία Donut (εικόνα 2.5) και ήρθε τον Σεπτέμβριο του 2009.



Εικόνα 2.5: Το λογότυπο του Android 1.6 Donut.

Πηγή: <http://logos.wikia.com>, Άρθρο *Android 1.6 Donut*

Το Donut είναι βασισμένο στον Linux Kernel 2.6.29. Στην έκδοση αυτή εμφανίζονται:

- Μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech)
- Εμφάνιση πληροφοριών χρήσης μπαταρίας
- Δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών ή αναζήτησης στο διαδίκτυο από την αρχική οθόνη
- Ανανεωμένο Android Market
- Βελτίωση της φωτογραφικής μηχανής με δυνατότητα εναλλαγής μεταξύ της λειτουργίας λήψης φωτογραφιών και βίντεο, καθώς και δυνατότητα διαγραφής πολλαπλών φωτογραφιών
- Υποστήριξη ανάλυσης οθόνων WVGA

Σε σύντομο χρονικό διάστημα, τον Οκτώβριο του 2009 βγήκε στην κυκλοφορία η επόμενη έκδοση Android 2.0, επονομαζόμενη Eclair (εικόνα 2.6), με δύο ενημερώσεις να ακολουθούν με υψηλότερο API τον Δεκέμβριο του 2009 (έκδοση Android 2.0.1 με API 6) και τον Ιανουάριο του 2010 (έκδοση Android 2.1 με API 7).



Εικόνα 2.6: Το λογότυπο του Android 2.0/2.1 Eclair.

Πηγή: <http://www.teleread.com>, Άρθρο *Android-Eclair*

Ομοίως με το Donut, βασισμένο στον Linux Kernel 2.6.29, αλλά με νέα χαρακτηριστικά όπως:

- Δυνατότητα προσθήκης πολλαπλών λογαριασμών Google σε μια συσκευή για συγχρονισμό των e-mail ή τηλεφωνικών καταλόγων
- Βελτιωμένες επιλογές επαφών
- Υποστήριξη Microsoft Exchange
- Δυνατότητα αναζήτησης στα SMS/MMS
- Νέες λειτουργίες της κάμερας όπως ψηφιακό ζουμ, επιλογή σκηνών, ισορροπία λευκού, ενσωματωμένο φλας, εστίαση macro
- Βελτιωμένη εμφάνιση και λειτουργικότητα εικονικού πληκτρολογίου
- Βελτιωμένος browser με υποστήριξη HTML5
- Υποστήριξη Bluetooth 2.1
- Εμφάνιση κινούμενων εικόνων φόντου (Live Wallpaper)
- Βελτιωμένο περιβάλλον χρήστη

Ακολουθεί η έκδοση του Android 2.2 τον Μάιο του 2010, με την ονομασία Froyo ως συντομογραφία του Frozen Yogurt (εικόνα 2.7). Το Froyo είχε 3 μικρές ενημερώσεις, αριθμημένες από 2.2.1 μέχρι 2.2.3 τον Νοέμβριο του 2011.



Εικόνα 2.7: Το λογότυπο του Android 2.2 Froyo.

Πηγή: <http://www.mobilexweb.com>, Άρθρο *Android 2.2 Froyo browser: HTML5, accelerometer, Flash Player 10.1 and more*

Βασισμένο στον Linux kernel 2.6.32, σαφώς γρηγορότερο από την προηγούμενη έκδοση, το Froyo περιλαμβάνει αλλαγές όπως:

- Εμφάνιση συντομεύσεων τηλεφώνου, λίστας εφαρμογών και browser στις 5 κύριες οθόνες για γρήγορη εκκίνηση
- Βελτιωμένη υποστήριξη Microsoft Exchange με νέες δυνατότητες όπως:
 - Απομακρυσμένη διαγραφή δεδομένων του κινητού σε περίπτωση κλοπής
 - Δυνατότητα κλειδώματος συσκευής με αριθμητικό ή αλφαριθμητικό pin
- Δυνατότητα ενεργοποίησης φλας κατά την εγγραφή βίντεο
- Πολλαπλές γλώσσες στο πληκτρολόγιο με απλή ενεργοποίηση των επιθυμητών και εναλλαγή αντίστοιχων λεξικών
- Δυνατότητα χρήσης του κινητού σαν WiFi Hotspot
- Ενσωμάτωση μηχανής V8 στον browser για γρηγορότερη πλοήγηση και φόρτωση σελίδων που χρησιμοποιούν Javascript

- Βελτίωση ταχύτητας επεξεργαστή από 2-5 φορές με τη χρήση του Dalvik JIT
- Δυνατότητα εναλλαγής αποθήκευσης εφαρμογών μεταξύ της μνήμης της συσκευής και της εξωτερικής κάρτας μνήμης
- Διεύρυνση δυνατοτήτων μέσω του Bluetooth
- Βελτίωση του Android Market
- Δυνατότητα αποστολής αρχείων στη συσκευή μέσω Cloud
- Υποστήριξη Adobe Flash 10.1

Ίσως η σημαντικότερη μέχρι στιγμής έκδοση είναι η επόμενη, καθώς μέχρι και σήμερα είναι από τις ευρέως χρησιμοποιούμενες. Ο λόγος για το Android 2.3, ή αλλιώς Gingerbread (εικόνα 2.8) το οποίο έκανε την εμφάνισή του τον Δεκέμβριο του 2010 αλλά ακολούθησε μια πληθώρα ενημερώσεων που κυμαίνονται από το Android 2.3.1 τον Δεκέμβριο του 2010, ως το 2.3.7 τον Μάιο του 2011. Στην έκδοση 2.3.3 που κυκλοφόρησε τον Φεβρουάριο του 2011 παρατηρείται αλλαγή του API από 9 σε 10.



Εικόνα 2.8: Το λογότυπο του Android 2.3-2.3.7 Gingerbread.

Πηγή: <http://www.brighthub.com>, Άρθρο *Preview of Android 2.3 Gingerbread*

Το Gingerbread είναι βασισμένο στην έκδοση Linux kernel 2.6.35. Ανάμεσα στις αλλαγές διακρίνονται:

- Βελτιωμένο περιβάλλον χρήστη για ευκολία και ταχύτητα
- Καινούρια σχεδίαση εικονικού πληκτρολογίου με διαισθητικό λεξικό
- Ευκολότερη αντιγραφή/επικόλληση με ένα άγγιγμα και προσδιορισμό κειμένου με κέρσορες
- Βελτιωμένη διαχείριση ενέργειας μπαταρίας
- Διαχείριση εφαρμογών με δυνατότητα τερματισμού αυτών
- Δυνατότητα κλήσης μέσω ίντερνετ με χρήστες λογαριασμών SIP
- Μετάδοση δεδομένων μέσω NFC
- Υποστήριξη δεύτερης κάμερας στη συσκευή, όπως της μπροστινής για βιντεοκλήσεις
- Εφαρμογή διαχείρισης λήψεων αρχείων
- Νέοι αισθητήρες, το γυροσκόπιο, το επιταχυνσιόμετρο, ο αισθητήρας περιστροφής, ο αισθητήρας επιτάχυνσης και το βαρόμετρο, χρησιμοποιούνται για τη δημιουργία 3D εφαρμογών
- Βελτιώσεις συστήματος για ταχύτερες εργασίες και παιχνίδια
- Δυνατότητα ρυθμίσεων ήχου

Τον Φεβρουάριο του 2011 και ενώ η προηγούμενη έκδοση ήταν ακόμα σε κυκλοφορία, βγήκε η επόμενη έκδοση, Android 3.0, με την ονομασία Honeycomb (εικόνα 2.9). Είναι η πρώτη έκδοση που είναι σχεδιασμένη για σε συσκευές με μεγαλύτερη οθόνη, συγκεκριμένα τα tablets. Οι ενημερώσεις του Honeycomb κυμαίνονται από την έκδοση Android 3.0 ως την έκδοση 3.2.6, τον Φεβρουάριο του 2012. Στις εκδόσεις 3.1 και 3.2 έχει ανέβει το επίπεδο του API σε 12 και 13 αντίστοιχα.



Εικόνα 2.9: Το λογότυπο του Android 3.0-3.2.6 Honeycomb.

Πηγή: <http://www.androidheadlines.com>, Άρθρο *Everything You Need to Know about Android 3.0 Honeycomb*

Το πρώτο tablet που κυκλοφόρησε με αυτή την έκδοση, βασιζόταν στον Linux kernel 2.6.36. Τα γραφικά ήταν αισθητά διαφορετικά σε σχέση με προηγούμενες εκδόσεις που προορίζονταν για κινητές συσκευές. Το περιβάλλον χρήστη χαρακτηρίζεται “ολογραφικό”. Οι αλλαγές περιλαμβάνουν:

- Γραμμή συστήματος στο κάτω μέρος της οθόνης με κουμπιά πλοήγησης
- Γραμμή ενεργειών στο πάνω μέρος της οθόνης, ενεργή ακόμα και όταν είναι ανοιχτή κάποια εφαρμογή
- Υποστήριξη πολυπύρηνων επεξεργαστών
- Υψηλές επιδόσεις γραφικών 2D και 3D
- Βελτιωμένο multitasking
- Δυνατότητα αλλαγής μεγέθους στα widget

- Ανανεωμένο πληκτρολόγιο
- Συνομιλία μέσω βίντεο με το Google Talk
- Δυνατότητα ιδιωτικής περιήγησης

Η επόμενη έκδοση έρχεται τον Οκτώβριο του 2011 και ονομάζεται Android 4.0, ή αλλιώς Ice Cream Sandwich (εικόνα 2.10). Πρόκειται για μια σημαντική έκδοση καθώς έρχεται με σκοπό να “ενώσει” κινητές συσκευές και tablet, παρέχοντας τους τη δυνατότητα να χρησιμοποιήσουν μια κοινή έκδοση με όμοιο περιβάλλον.

Οι ενημερώσεις του Ice Cream Sandwich, ή ICS όπως αναφέρεται συχνά για συντομία, ξεκινούν από το Android 4.0 και φτάνουν το 4.0.4 τον Μάρτιο του 2012 με αναβάθμιση του API στην έκδοση 4.0.3.



Εικόνα 2.10: Το λογότυπο του Android 4.0-4.0.4 Ice Cream Sandwich.

Πηγή: <http://www.huffingtonpost.com>, Άρθρο *Google Android Ice Cream Sandwich OS And Logo Get Official*

Βασισμένο στον Linux Kernel 3.0.1, χρησιμοποιεί στοιχεία του Gingerbread και του Honeycomb για να δημιουργήσει ένα περιβάλλον κατάλληλο για τους χρήστες οποιασδήποτε συσκευής. Μεταξύ των αλλαγών εμφανίζονται:

- Νέα γραμματοσειρά στο περιβάλλον χρήστη
- Ξεκλείδωμα συσκευής μέσω αναγνώρισης προσώπου
- Δυνατότητα άμεσης μετάβασης στην κάμερα από την κλειδωμένη οθόνη

- Γρήγορες απαντήσεις για τις εισερχόμενες κλήσεις με προσαρμοσμένα γραπτά μηνύματα
- Κλείσιμο ανεπιθύμητων εφαρμογών, καρτελών και ενημερώσεων με ένα άγγιγμα
- Βελτιωμένος browser με δυνατότητα συγχρονισμού σελιδοδεικτών μέσω του Chrome sync με τον σταθερό υπολογιστή
- Ενσωματωμένος μετρητής δεδομένων
- Νέες δυνατότητες στη λήψη φωτογραφιών και βίντεο
- Υποστήριξη WiFi peer-to-peer (P2P)
- Υποστήριξη Bluetooth Health Device Profile (HDP)
- Βελτίωση λειτουργίας φωνητικών εντολών

Τον Ιούλιο του 2012 κάνει την εμφάνισή του το Android 4.1, ή αλλιώς Jelly Bean (εικόνα 2.11). Είναι η πρώτη από τις συνολικά 3 εκδόσεις του Jelly Bean, ακολουθούν οι ενημερώσεις 4.1.1, 4.1.2, έπειτα η έκδοση 4.2 με ενημερώσεις ως την 4.2.2 και τέλος η 4.3 τον Ιούλιο του 2013. Το πρότυπο “μία έκδοση για όλες τις συσκευές” συνεχίζεται και εξελίσσεται στο Jelly Bean. Η έκδοση αυτή έχει ένα νέο περιβάλλον χρήστη για tablet με μικρή οθόνη. Κάτι που δεν έχει ξανασυμβεί σε καμία από τις προηγούμενες εκδόσεις του Android είναι ότι παρόλο που μπορεί να γινόταν αναβάθμιση του API μέσα σε μια ονομαστικά ίδια έκδοση, εξακολουθούσε να βασίζεται στον ίδιο Linux kernel. Οι 3 εκδόσεις του Jelly Bean βασίζονται η κάθε μία σε διαφορετικό. Η έκδοση 4.1.x (όπου x ο αριθμός των ενημερώσεων που ακολουθούν, για συντομία) βασίζεται στον Linux kernel 3.0.31, η έκδοση 4.2.x στον Linux kernel 3.4.0 και η 4.3 στον Linux kernel 3.4.39.



Εικόνα 2.11: Το λογότυπο του Android 4.1-4.3 Jelly Bean.

Πηγή: <http://www.technesstivity.com>, Άρθρο *Android 4.1: What's New In Jelly Bean?*

Το Jelly Bean φέρει νέα χαρακτηριστικά όπως:

- Νέα σχεδίαση ειδοποιήσεων
- Μεγαλύτερη απόδοση
- Αναζήτηση μέσω του Google Now
- Δυνατότητα φωνητικής αναζήτησης
- Πληκτρολόγιο με λειτουργία πρόγνωσης των επόμενων λέξεων
- Βελτίωση ξεκλειδώματος συσκευής μέσω αναγνώρισης προσώπου με έλεγχο ζωτικότητας
- Φωνητικές εντολές χωρίς απαραίτητη σύνδεση στο internet
- Μετάδοση δεδομένων μέσω του Android Beam

- Δυνατότητα πολλαπλών χρηστών σε tablets χωρίς αναγκαστική αποσύνδεση
- Εμφάνιση widgets στην κλειδωμένη οθόνη
- Προστασία οθόνης με φωτογραφίες του χρήστη
- Δυνατότητα λήψης εικόνων 360 μοιρών με το Photo Sphere
- Υποστήριξη γλωσσών γραμμένων από δεξιά προς τα αριστερά

Η πιο πρόσφατη, μέχρι στιγμής, έκδοση λογισμικού είναι η επόμενη και πρόκειται για το Android 4.4 με την ονομασία KitKat (εικόνα 2.12). Έκανε την εμφάνισή του τον Οκτώβριο του 2013 ενώ έχουν ήδη κυκλοφορήσει 2 ενημερώσεις, η 4.4.1 και η 4.4.2 τον Δεκέμβριο του ίδιου έτους.



Εικόνα 2.12: Το λογότυπο του Android 4.4, KitKat.

Πηγή: <http://www.android.com>, Άρθρο *KitKat 4.4*

Προτού κυκλοφορήσει, ο κόσμος πίστευε ότι αυτή η έκδοση θα είχε τον αριθμό 5.0 και θα ονομαζόταν Key Lime Pie. Τελικά προς έκπληξη όλων ονομάστηκε KitKat. Η έκδοση αυτή στοχεύει στην αποτελεσματικότητα σε συνδυασμό με την χρηστικότητα, καθώς μπορεί να λειτουργήσει ακόμα και με μνήμη RAM 512MB.

Οι αλλαγές που φέρει είναι:

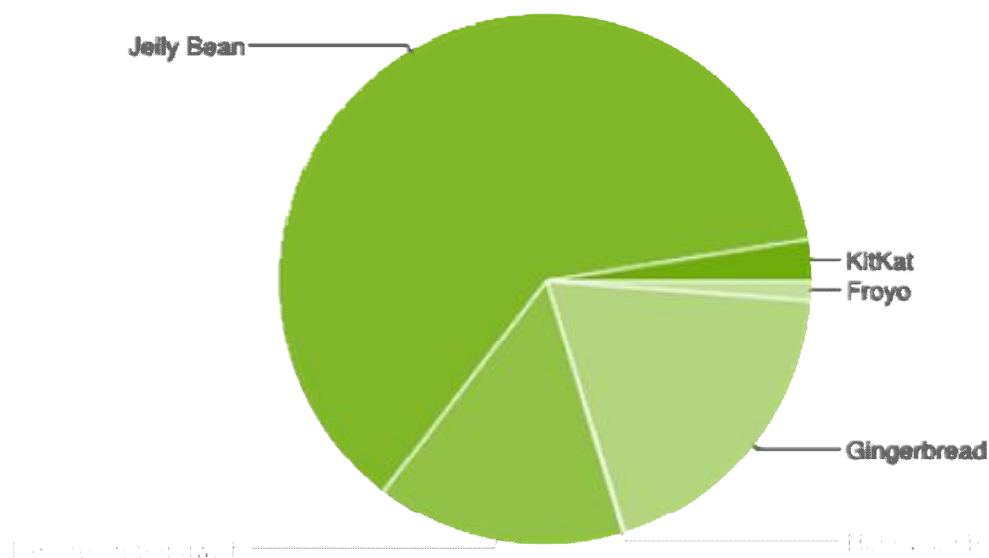
- Συναλλαγές με χρήση του NFC μέσω του Host Card Emulation (HCE)
- Βελτίωση του Google Now
- Απομακρυσμένη εκτύπωση εγγράφων μέσω του Google Cloud Print
- Βελτίωση απόδοσης
- Υποστήριξη 2 νέων αισθητήρων, καταμετρητή και εντοπισμού βημάτων.
- Λειτουργία “έξυπνης” αναγνώρισης κλήσης με λήψη στοιχείων του καλούντος μέσω του προφίλ του στο Google Maps (αν είναι διαθέσιμο)
- Υποστήριξη καταγραφής της οθόνης και των κινήσεων σε βίντεο (screen recording)
- Εύκολη πρόσβαση σε αρχεία που βρίσκονται σε ιστοσελίδες τύπου Cloud
- Υποστήριξη εμφάνισης εφαρμογών full screen
- Υποστήριξη νέων λειτουργιών μέσω Bluetooth
- Εντοπισμός συσκευής μέσω του Android Device Manager

2.8 Ποσοστά χρήσης των εκδόσεων Android

Είναι ενδιαφέρουσα η πρόσφατη έρευνα της εταιρείας Android για το ποσοστό εγκατεστημένων εκδόσεων σε συσκευές Android στην παρούσα χρονική περίοδο. Η έρευνα αυτή διήρκεσε 7 ημέρες, με ημερομηνία λήξης την 3^η Μαρτίου 2014 και καταμέτρησε τα ποσοστά εκδόσεων που είναι εγκατεστημένες σε όλες τις συσκευές Android (πίνακας 2.1). Στον πίνακα καταγράφονται αναλυτικά και όλα τα επίπεδα API ανά έκδοση. Να σημειωθεί ότι εκδόσεις με ποσοστό χρήσης κάτω του 0,1% δεν συμπεριλαμβάνονται στο σχήμα 2.2.

ΕΚΔΟΣΗ	ΟΝΟΜΑΣΙΑ	API	ΚΑΤΑΝΟΜΗ
Android 1.0		1	
Android 1.1		2	
Android 1.5	Cupcake	3	
Android 1.6	Donut	4	
Android 2.0		5	
Android 2.0.1		6	
Android 2.1	Eclair	7	
Android 2.2-2.2.3	Froyo	8	1.2%
Android 2.3-2.3.2		9	
Android 2.3.3-2.3.7	Gingerbread	10	19.0%
Android 3.0		11	
Android 3.1		12	
Android 3.2	Honeycomb	13	0.1%
Android 4.0-4.0.2		14	
Android 4.0.3-4.0.4	Ice Cream Sandwich	15	15.2%
Android 4.1-4.1.2		16	35.3%
Android 4.2-4.2.2		17	17.1%
Android 4.3	Jelly Bean	18	9.6%
Android 4.4	KitKat	19	2.5%

Πίνακας 2.1: Ποσοστό εγκατεστημένων εκδόσεων σε συσκευές Android.
 Πηγή: <https://developer.android.com>, ο Πίνακας σχεδιάστηκε με χρήση των πληροφοριών του Dashboard στις 17 Μαρτίου 2014



Σχήμα 2.2: Πίτα εγκατεστημένων εκδόσεων σε συσκευές Android.
 Πηγή: <https://developer.android.com>, Άρθρο *Dashboards* στις 17 Μαρτίου 2014

2.9 Επιλογή έκδοσης για την εφαρμογή

Οι εφαρμογές που χρησιμοποιούν παλαιότερες εκδόσεις λογισμικού λειτουργούν συνήθως χωρίς κανένα πρόβλημα σε συσκευές με νεότερες εκδόσεις. Με αυτό σαν δεδομένο, λόγω της μεγάλης χρήσης της έκδοσης Gingerbread ακόμη και σήμερα, αποφασίστηκε η έκδοση που θα χρησιμοποιηθεί στην εφαρμογή να είναι η Android 2.3.3 Gingerbread.

2.10 Άλλα λειτουργικά συστήματα κινητών συσκευών

Ένα μερίδιο της αγοράς χρησιμοποιεί συσκευές με διαφορετικό λειτουργικό σύστημα από αυτό του Android. Άξια αναφοράς είναι τα γνωστότερα 4, εκ των οποίων μερικά υπήρχαν πολύ πριν την εμφάνιση του Android και υπήρχαν στις περισσότερες κινητές συσκευές της εποχής εκείνης. Ο λόγος για το iOS, το Blackberry OS, το Symbian και το Windows Mobile.

2.10.1 iOS

Το λειτουργικό σύστημα iOS ανήκει στην εταιρεία Apple και έκανε την εμφάνισή του το 2007, αποτελώντας το λειτουργικό σύστημα για το iPhone, κυρίως, καθώς και για τις υπόλοιπες συσκευές της σειράς παραγωγής (iPod touch, iPad). Αποτελεί τον έναν εκ των τριών κύριων ανταγωνιστών στην σημερινή αγορά λειτουργικών συστημάτων κινητών συσκευών τηλεφωνίας.

Σε αντίθεση ωστόσο με το Android της Google και τα Windows της Microsoft, δεν διατίθεται προς χρήση σε συσκευές τρίτων εταιρειών. Αυστηρά κλειστής αρχιτεκτονικής έφτασε να κατέχει ακόμα και 60% επί του συνόλου των λειτουργικών συστημάτων. Το ποσοστό αυτό μετέπειτα μειώθηκε πολύ λόγω της ανόδου του Android.

Αξίζει να σημειωθεί πως σε αντίθεση ακόμα και με παλαιότερα λειτουργικά, για παράδειγμα το Symbian, το iOS απέκτησε πραγματικό multitasking από την έκδοση 4 και μετά. Έως τότε περιορίζονταν σε ορισμένες μόνο εφαρμογές, ένας περιορισμός που είχε επιβληθεί επί σκοπού από την Apple.

2.10.2 Blackberry OS

Το λειτουργικό σύστημα Blackberry OS εισήλθε στην αγορά κινητής τηλεφωνίας με την πρώτη του έκδοση τον Μάρτιο του 2002. Κατασκευασμένο από την

Blackberry Limited για την ομώνυμη σειρά κινητών αποτέλεσε ένα από τα πρώτα λειτουργικά συστήματα κινητών συσκευών με ενσωματωμένες δυνατότητες multitasking. Διέθετε πλήρες σετ υποστήριξης εφαρμογών πολυμέσων, καθώς και μια πληθώρα συσκευών εισόδου κατασκευασμένες από την ίδια την Blackberry, όπως trackball και οθόνη αφής.

Κατέστη ιδιαίτερα δημοφιλές στον επιχειρηματικό κόσμο κυρίως λόγω της εφαρμογής email που ήταν ειδικά διαμορφωμένη για επιχειρήσεις. Το 2011 έφτασε να κατέχει μερίδιο αγοράς της τάξης του 45%. Το διαδέχθηκε το 2013 το Blackberry 10.

2.10.3 Windows Mobile

Το λειτουργικό Windows Mobile βασίζεται στον πυρήνα των Windows CE (Windows Embedded Compact). Η πλειοψηφία των εκδόσεων του διαθέτουν εγγενείς λειτουργίες multitasking καθώς και δυνατότητες διαμοιρασμού της εκάστοτε σύνδεσης στο διαδίκτυο με τρίτες συσκευές, είτε μέσω USB, είτε μέσω Bluetooth.

Κυρίαρχο κομμάτι του αποτελεί η Home Screen (παλαιότερα ονομαζόμενη Today Screen). Υποστηρίζεται ακόμα η εκτέλεση εφαρμογών από τρίτους κατασκευαστές, κάτι που αποτέλεσε χαρακτηριστική του λειτουργία σχεδόν από την γέννηση του. Το μέγιστο ποσοστό διείσδυσης στην αγορά κινητής τηλεφωνίας που πέτυχε ήταν της τάξης του 42% κατά το έτος 2007. Έκτοτε, όπως και το Blackberry OS, γνώρισε τεράστια πτώση για να σταθεροποιηθεί τελικά σε μονοψήφια νούμερα.

2.10.4 Symbian

Αποτέλεσε ένα από τα ευρύτερα διαδεδομένα λειτουργικά κινητών τηλεφώνων στα τέλη της δεκαετίας του 90 και για αρκετά χρόνια μετέπειτα. Το 2010, λίγο πριν εκθρονιστεί από το Android, κυριαρχούσε παγκοσμίως. Μεγάλο κομμάτι της επιτυχίας του προήλθε από την χρήση του στην πλατφόρμα S60 της Nokia. Υποστήριζε εγγενώς πολλαπλές γλώσσες, ανάμεσα τους και τα Ελληνικά. Έδινε ιδιαίτερη έμφαση στην αυξημένη συμβατότητα με εξωτερικές συσκευές αποθήκευσης δεδομένων. Το περιβάλλον του σε μεγάλο βαθμό βασίστηκε σε παραμετροποιημένες εκδόσεις τρίτων κατασκευαστών, με μόνο τα πολύ βασικά στοιχεία και υποστοιχεία να βρίσκονται εντός του κεντρικού του πυρήνα.

Αδυνατώντας να προσαρμοστεί στις σύγχρονες απαιτήσεις των κινητών που βασίζονται στις οθόνες αφής πέρασε στο περιθώριο καθώς πλέον χρησιμοποιείται από ελάχιστες συσκευές.

ΚΕΦΑΛΑΙΟ 3

Σχεδιασμός και Ανάλυση Εφαρμογής

3.1 Εισαγωγή

Για τη δημιουργία μιας εφαρμογής Android απαιτείται μια σειρά ενεργειών. Για αρχή φυσικά χρειάζεται μια ιδέα για εφαρμογή και πρέπει να γίνει κατανοητό το πώς λειτουργεί. Για την κατασκευή αυτής, χρειάζονται τα κατάλληλα εργαλεία στη μορφή προγραμμάτων. Είναι πολύ σημαντικό επίσης, να γίνει ο σχεδιασμός για το τι θα περιέχει και να παραχθούν οι βάσεις πριν την υλοποίηση.

3.2 Αρχική ιδέα μιας εφαρμογής

Για τη δημιουργία μιας πρωτότυπης εφαρμογής απαιτείται μελέτη και φαντασία. Με την πάροδο του χρόνου, όλο και περισσότερες εφαρμογές κάνουν την εμφάνισή τους λόγω της δωρεάν παροχής εργαλείων ανάπτυξης και του μεγάλου ανταγωνισμού. Έτσι, ακόμα και η σκέψη κάποιας πρωτοτυπίας είναι δύσκολη. Μελετώντας τι λείπει από το Android Market μπορεί κανείς να παρατηρήσει που υπάρχουν κενά και να προσπαθήσει να τα καλύψει. Κάπως έτσι προέκυψε και η ιδέα αυτής της εφαρμογής.

3.3 Στόχος της εφαρμογής

Ο στόχος της εφαρμογής είναι να λειτουργήσει σαν εργαλείο στα χέρια των κατοίκων μιας πόλης, ώστε να μπορούν να ενημερώσουν τους υπεύθυνους στο Δήμο για τα προβλήματα που συναντούν και να επιδιορθώνονται άμεσα. Με τη αξιοποίηση των αισθητήρων της κάμερας και του GPS, η εφαρμογή θα συλλέγει στοιχεία από την τρέχουσα θέση του χρήστη, μια φωτογραφία απεικόνισης του προβλήματος που υφίσταται και γραπτές σημειώσεις με επιπλέον πληροφορίες που θα μπορούν να βοηθήσουν. Έπειτα, ο χρήστης αποστέλλει το αίτημά του μέσω email στο Δήμο, ο οποίος με τη σειρά του καλείται να ελέγξει και να διορθώσει τις βλάβες. Το όνομα που επιλέχθηκε για την εφαρμογή είναι Citizen Call.

3.4 Εγκατάσταση λογισμικού

Για την υλοποίηση της αρχικής ιδέας μιας εφαρμογής, ο προγραμματιστής θα χρειαστεί μια σειρά εργαλείων που θα τον βοηθήσουν στην ανάπτυξη, τον έλεγχο, το σχεδιασμό, τη δοκιμή και την επιδιόρθωση αυτής. Απαραίτητο είναι αρχικά ένα περιβάλλον ανάπτυξης της εφαρμογής, το οποίο ονομάζεται και Ολοκληρωμένο Περιβάλλον Ανάπτυξης (Integrated Development Environment – IDE). Ένα IDE είναι μια εφαρμογή λογισμικού που παρέχει ολοκληρωμένες υπηρεσίες στον προγραμματιστή για την ανάπτυξη άλλων λογισμικών. Αποτελείται συνήθως από ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα, εργαλεία για compile και διεργασίες, ενώ παρέχει και αρκετές διευκολύνσεις όπως για παράδειγμα αυτόματη συμπλήρωση κώδικα και ανάλογος χρωματισμός σε κάθε εντολή.

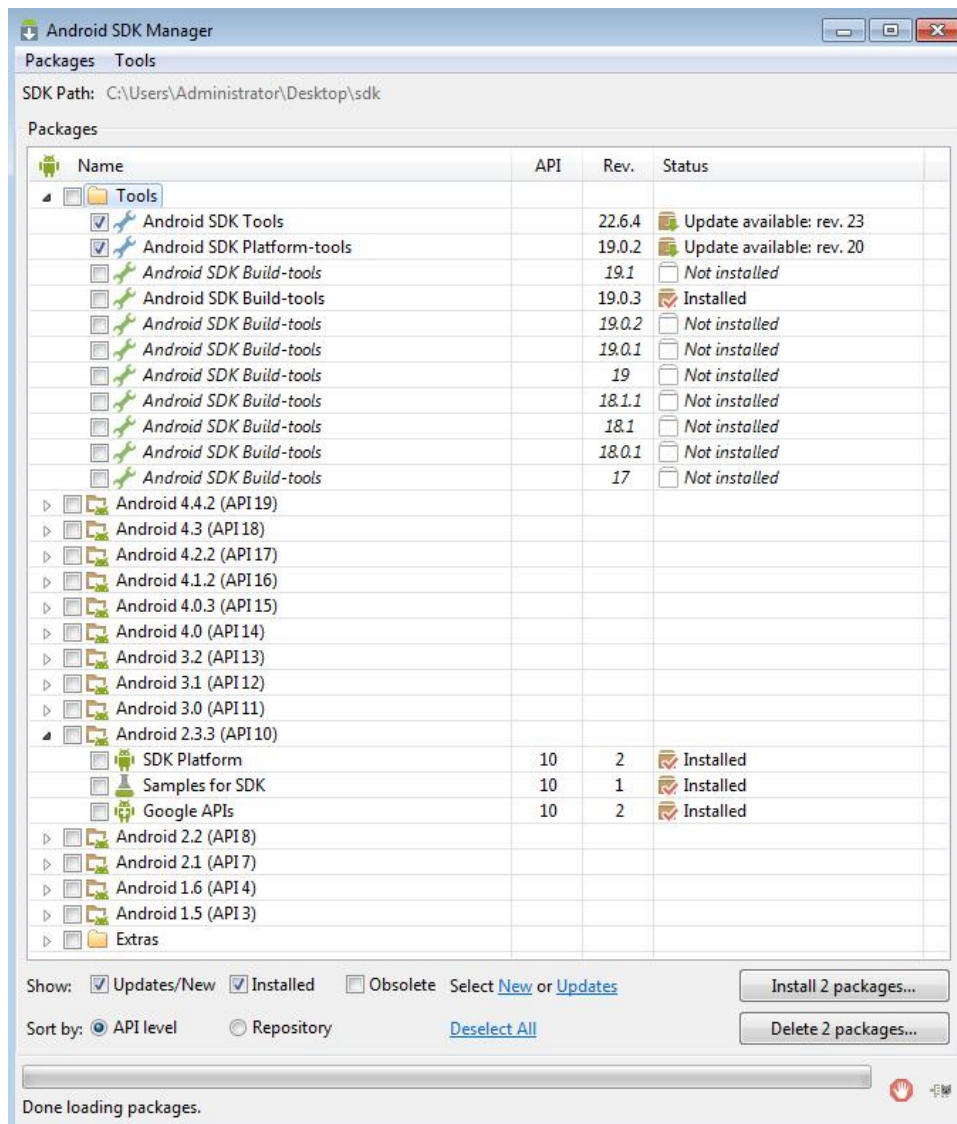
Επιπλέον, ο προγραμματιστής θα χρειαστεί και το Android Software Development Kit (SDK), το λογισμικό που επιτρέπει τη δημιουργία εφαρμογών για την πλατφόρμα του Android.

3.4.1 Android SDK

Το λογισμικό Android SDK διανέμεται δωρεάν μέσα από την ιστοσελίδα του Android και είναι απαραίτητο για την ανάπτυξη μιας εφαρμογής, καθώς περιέχει όλες τις αναγκαίες πληροφορίες. Κατά το κατέβασμα ο προγραμματιστής λαμβάνει ένα συμπίεμένο αρχείο zip με τα περιεχόμενα του SDK. Μετά την αποσυμπίεση δεν χρειάζεται κάποια εγκατάσταση προγράμματος. Υπάρχουν δυο εφαρμογές διαθέσιμες μέσα στο φάκελο του SDK, το SDK Manager και το AVD Manager. Οι εφαρμογές Android είναι γραμμένες σε γλώσσα προγραμματισμού Java. Τα εργαλεία του Android SDK παρέχουν τη δυνατότητα του compile του κώδικα και όλων των αρχείων και φακέλων, σε ένα αρχείο APK (Android Package, με κατάληξη .apk). Το αρχείο αυτό περιέχει όλες τις πληροφορίες του κώδικα και χρησιμοποιείται για να γίνει εγκατάσταση της εφαρμογής σε μια κινητή συσκευή Android.

Το SDK Manager είναι ο διαχειριστής των εργαλείων και των εκδόσεων που μπορεί να κατεβάσει ο προγραμματιστής (εικόνα 3.1). Για εξοικονόμηση χώρου, κατά τη λήψη του SDK δεν συμπεριλαμβάνονται παρά μόνο τα απαραίτητα εργαλεία και η τελευταία έκδοση του Android. Ο προγραμματιστής έχει την επιλογή να κατεβάσει παλαιότερες εκδόσεις εάν το επιθυμεί. Για την εφαρμογή

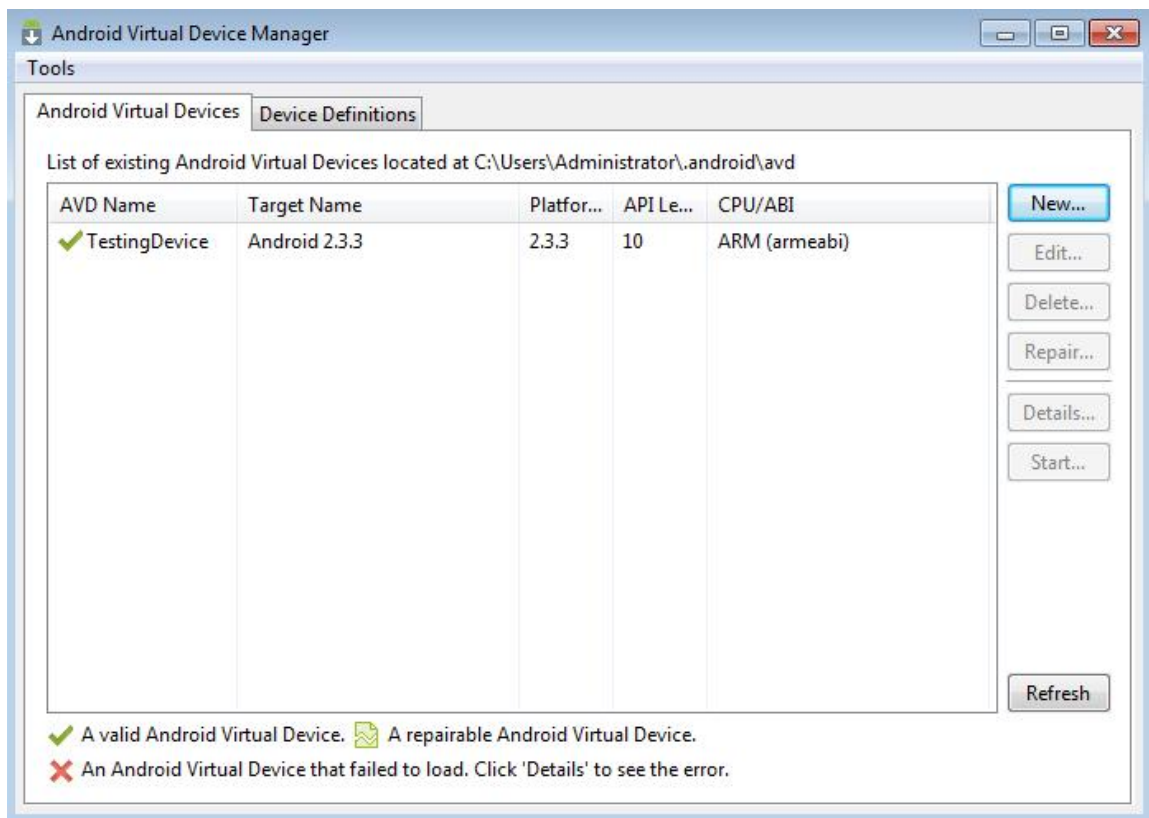
έχει χρησιμοποιηθεί η Android 2.3.3 ως η παλαιότερη διαθέσιμη έκδοση, οπότε για την ανάπτυξη είναι απαραίτητη η ύπαρξή της. Το SDK περιέχει έτοιμα προγράμματα σαν δείγματα ώστε να έχει κάποιος αρχάριος μια πρώτη εικόνα του πηγαίου κώδικα και μιας εφαρμογής, εργαλεία ανάπτυξης και debugging, έναν εξομοιωτή (emulator), compiler, και απαραίτητες βιβλιοθήκες για την ανάπτυξη εφαρμογών Android.



Εικόνα 3.1: SDK Manager.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Android SDK Manager

Το AVD Manager (Android Virtual Device) είναι ο διαχειριστής των εικονικών συσκευών που χρησιμοποιούνται ως εξομοιωτές της εκάστοτε εφαρμογής (εικόνα 3.2).



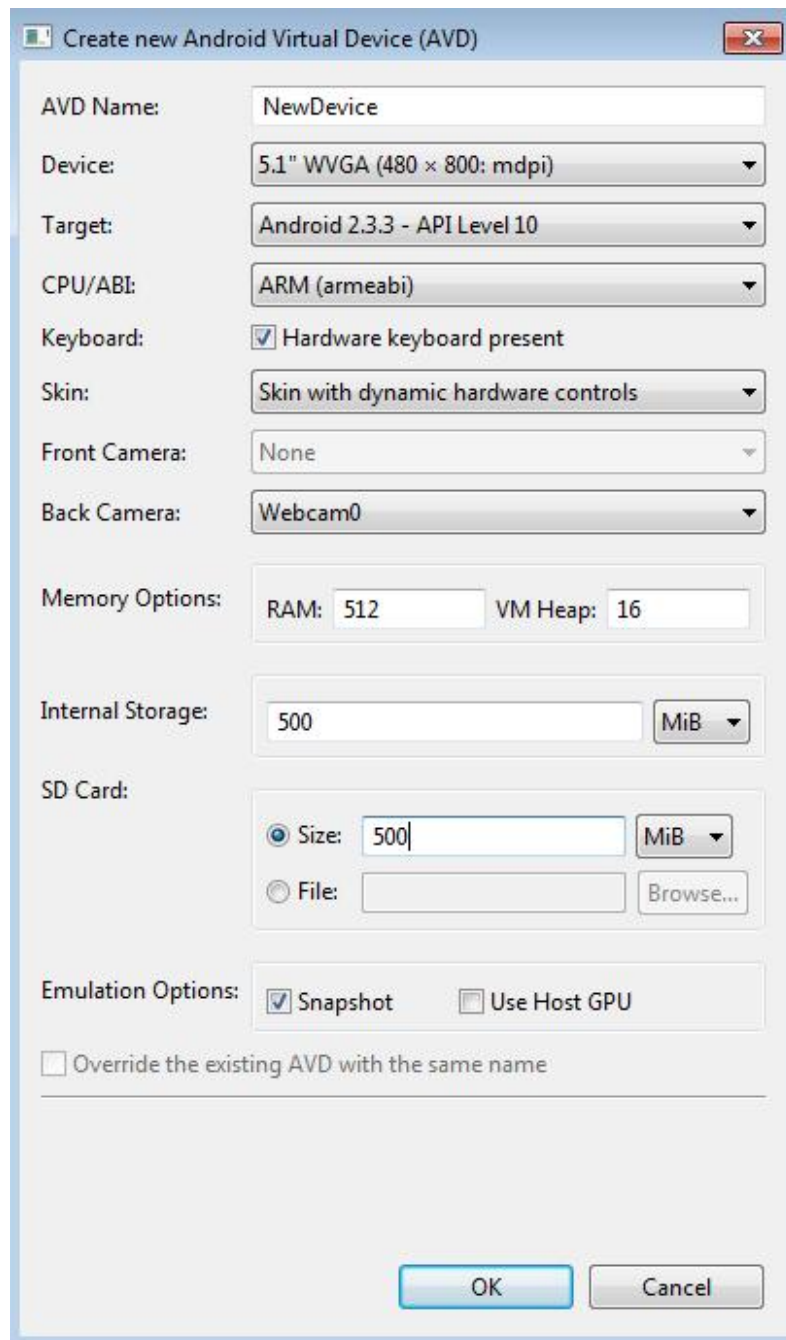
Εικόνα 3.2: AVD Manager.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του AVD Manager

Ο προγραμματιστής καλείται να δημιουργήσει μια καινούρια εικονική συσκευή, της οποίας τα χαρακτηριστικά μπορεί να προσαρμόσει όπως αυτός επιθυμεί (εικόνα 3.3). Δίνονται ως προεπιλογές οι συσκευές Nexus με αυτόματη συμπλήρωση των χαρακτηριστικών της εσωτερικής μνήμης τους και της RAM, ειδάλως υπάρχει μια πληθώρα μεγεθών οθόνης για να διαλέξει ο προγραμματιστής και χειροκίνητη συμπλήρωση των υπόλοιπων πεδίων. Τα χαρακτηριστικά μιας συσκευής που μπορούν να οριστούν στον εξομοιωτή είναι:

- Η συσκευή (Device), όπου δίνονται σαν επιλογές όλα τα διαθέσιμα μεγέθη οθονών με πληροφορίες ανάλυσης ή οι συσκευές Nexus με όλα τα χαρακτηριστικά τους προσαρτημένα.
- Η έκδοση Android (Target) που θα χρησιμοποιεί ο εξομοιωτής.
- Η έκδοση της CPU (CPU/ABI) που θα έχει ο εξομοιωτής.

- Το πληκτρολόγιο (Keyboard), όπου δίνεται η επιλογή χρήσης του πληκτρολογίου του υπολογιστή για την πληκτρολόγηση εντός της εξομοίωσης.
- Η εμφάνιση του εξομοιωτή (Skin), η κινητή συσκευή που αντιπροσωπεύει εξωτερικά.
- Οι κάμερες (Front και Back Camera), όπου δίνεται η δυνατότητα εξομοίωσης μιας κάμερας ή χρήσης μιας πραγματικής εφόσον ο υπολογιστής διαθέτει μια.
- Επιλογές μνήμης (Memory Options), όπου ορίζεται η μνήμη RAM της συσκευής.
- Εσωτερική μνήμη (Internal Storage), για τον ορισμό της εσωτερικής μνήμης της συσκευής.
- Κάρτα SD (SD Card), για τον ορισμό της εξωτερικής μνήμης της συσκευής.
- Επιλογές Εξομοίωσης (Emulation Options), με επιλογές Εικόνας (Snapshot) και Χρήση Κάρτας Γραφικών του Χρήστη (Use Host GPU)
 - Το Snapshot χρησιμοποιείται για να επιταχύνει την εκκίνηση του εξομοιωτή. Επιλέγοντάς το, το σύστημα κρατάει μια εικόνα της κατάστασης της μνήμης RAM του εξομοιωτή όταν έχει ανοίξει. Με τη χρήση αυτής της εικόνας επαναφέρει τον εξομοιωτή πιο γρήγορα όταν ξανανοίξει.
 - Το Use Host GPU χρησιμοποιεί την εφαρμογή OpenGL του υπολογιστή για να αξιολογήσει τις εντολές μέσα στο περιβάλλον εξομοίωσης. Με αυτό τον τρόπο, αυξάνεται δραματικά η ταχύτητα του εξομοιωτή.

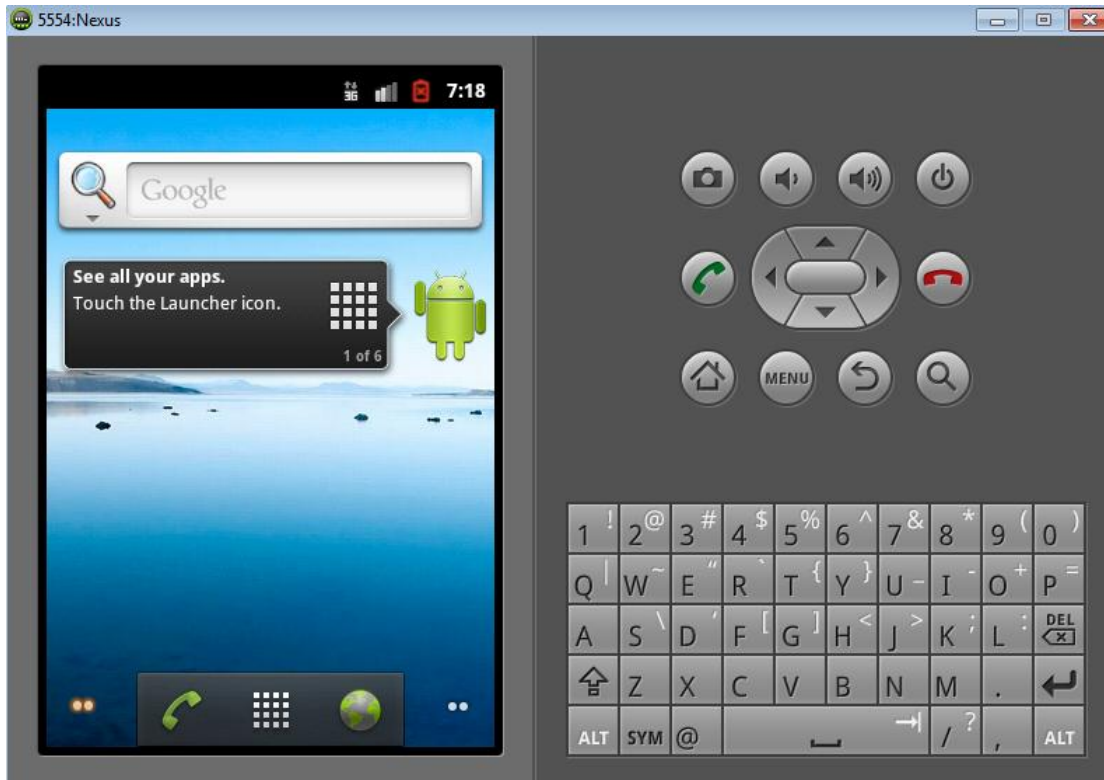


Εικόνα 3.3: Δημιουργία νέας εικονικής συσκευής.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του AVD Manager

Οι εικονικές συσκευές εξυπηρετούν στην δοκιμασία μιας εφαρμογής πριν την δημοσίευσή της. Για τη σωστή λειτουργία θα πρέπει ο προγραμματιστής να την δοκιμάσει σε μια ποικιλία διαφορετικών συσκευών και εκδόσεων. Η φυσική απόκτηση πολλαπλών συσκευών όμως κοστίζει ακριβά. Αυτό το πρόβλημα έρχεται να λύσει η εξομοίωση καθώς η δημιουργία εικονικών συσκευών είναι μια απλή και γρήγορη διαδικασία με τη χρήση του AVD Manager.

Το αποτέλεσμα της εξομοίωσης είναι σχεδόν ίδιο με αυτό της πραγματικής συσκευής, καθώς ο εξομοιωτής χρησιμοποιεί τα αντίστοιχα χαρακτηριστικά οθόνης και API (εικόνα 3.4). Υστερεί σε σχέση με τις φυσικές συσκευές στο ότι πιθανώς να διαφέρει στο περιεχόμενο του συστήματος.



Εικόνα 3.4: Το περιβάλλον του εξομοιωτή.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του AVD Simulator

3.4.2 Eclipse

Το Eclipse είναι ένα λογισμικό IDE που υποστηρίζει Java και παρέχεται δωρεάν μαζί με το Android SDK από την ιστοσελίδα του, με ενσωματωμένα τα Εργαλεία Ανάπτυξης του Android (Android Development Tools – ADT). Παλαιότερα, ένας προγραμματιστής καλούνταν να εγκαταστήσει χειροκίνητα το plugin των ADT, αλλά καθώς το Eclipse ήταν το δημοφιλέστερο IDE που χρησιμοποιούνταν για την ανάπτυξη εφαρμογών Android, πλέον παρέχεται έτοιμο. Δεν απαιτείται εγκατάσταση για να λειτουργήσει, πράγμα που σημαίνει ότι είναι έτοιμο προς χρήση από τη στιγμή που το έχει κατεβάσει κάποιος από το internet. Εναλλακτικά μπορεί να χρησιμοποιηθεί οποιοσδήποτε άλλος Text Editor ή λογισμικό IDE που να υποστηρίζει Java για την ανάπτυξη εφαρμογών Android.

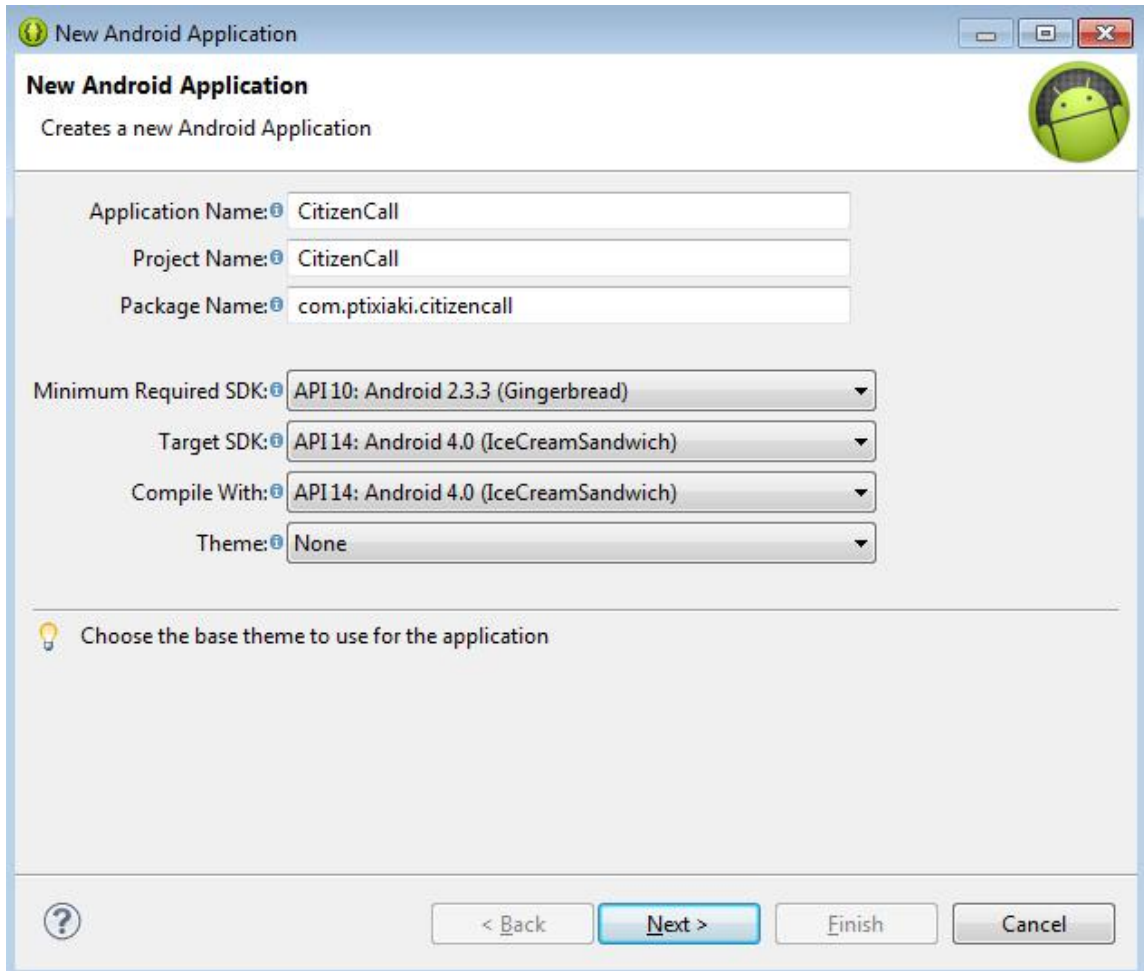
Κατά την εκκίνηση ο προγραμματιστής ζητείται να ορίσει την τοποθεσία που θα δημιουργηθεί ο φάκελος workspace, όπου θα αποθηκεύονται τα μελλοντικά project του Eclipse.

Για την ανάπτυξη μιας εφαρμογής Android απαιτείται η δημιουργία ενός νέου Android Application Project. Για την ενέργεια αυτή, ακολουθείται το path File – New – Android Application Project. Ο προγραμματιστής καλείται να συμπληρώσει τα χαρακτηριστικά της νέας του εφαρμογής (εικόνα 3.5). Τα πεδία περιλαμβάνουν:

- Το όνομα της εφαρμογής (Application Name), το οποίο θα εμφανίζεται στο Play Store και στη συσκευή όπου θα χρησιμοποιείται.
- Το όνομα του project (Project Name), το οποίο θα χρησιμοποιείται αποκλειστικά από το Eclipse για την ονομασία του φακέλου του project στο workspace. Ενδείκνυται η χρήση του ίδιου ονόματος με αυτό της εφαρμογής για ευκολία.
- Το όνομα του πακέτου (Package Name), το οποίο είναι συνήθως η αντίστροφη διεύθυνση της ιστοσελίδας του προγραμματιστή (εάν αυτός διαθέτει κάποια), μαζί με μια αναγνωριστική ονομασία της εφαρμογής. Το όνομα του πακέτου δεν εμφανίζεται συνήθως στους χρήστες της εφαρμογής, και πρέπει να παραμείνει το ίδιο καθ' όλη τη διάρκεια της εφαρμογής διότι είναι το κλειδί που αναγνωρίζει πολλαπλές εκδόσεις μιας εφαρμογής ως την ίδια εφαρμογή.
- Την μικρότερη έκδοση Android (Minimum Required SDK), όπου ορίζεται η παλαιότερη έκδοση Android την οποία θα υποστηρίζει η εφαρμογή. Για το Citizen Call έχει επιλεγεί η Android 2.3.3 Gingerbread, καθώς βάσει των ερευνών χρησιμοποιείται ακόμη από ένα μεγάλο μερίδιο των χρηστών.
- Την έκδοση Android που στοχεύει (Target SDK), δηλαδή την υψηλότερη έκδοση Android που θα υποστηρίζει η εφαρμογή. Αυτό το πεδίο ενημερώνει το σύστημα ότι η εφαρμογή έχει δοκιμαστεί στην συγκεκριμένη έκδοση και δεν έχει εμφανίσει προβλήματα συμβατότητας. Η εφαρμογή επίσης θα λειτουργεί στις ενδιάμεσες εκδόσεις ανάμεσα στην παλαιότερη που έχει επιλεγεί και σε αυτήν.
- Μεταγλώττιση (Compile With), πιο συχνά αναφερόμενο απλά ως compile. Σε αυτό το πεδίο ο προγραμματιστής επιλέγει την έκδοση στην οποία

επιθυμεί να γίνει το compile. Συνήθως είναι η πιο πρόσφατη έκδοση η οποία υποστηρίζει τα περισσότερα API, για αποφυγή προβλημάτων.

- Θέμα (Theme), όπου ο προγραμματιστής μπορεί να επιλέξει από μια μικρή λίστα ένα θέμα για την εμφάνιση της εφαρμογής του.



Εικόνα 3.5: Δημιουργία νέου project εφαρμογής στο Eclipse.

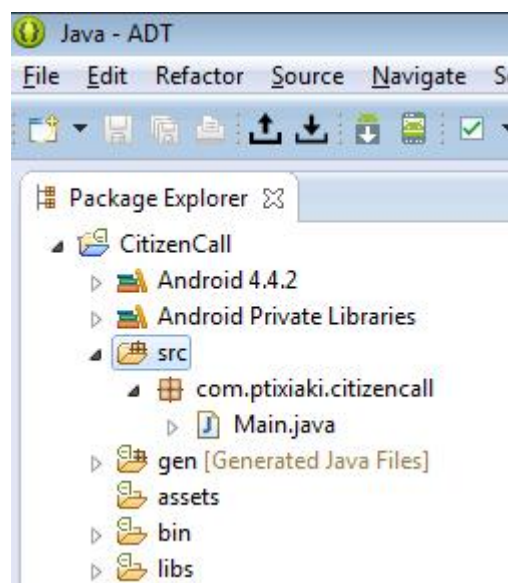
Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, New Project

Στη συνέχεια ο προγραμματιστής καλείται να ορίσει σε ποιο workspace επιθυμεί να δημιουργηθεί το νέο project, εάν επιθυμεί κάποιο δικό του εικονίδιο για την εφαρμογή (το πεδίο αυτό μπορεί να το αλλάξει και αργότερα), και καλείται να ορίσει τη νέα δραστηριότητα (activity) από την οποία θα ξεκινήσει.

3.5 Το εσωτερικό μιας εφαρμογής

Μια εφαρμογή αποτελείται από ένα σύνολο φακέλων και αρχείων τα οποία συντελούν ένα project μέσα στο Eclipse. Στην εκκίνηση ενός νέου project, το Eclipse δημιουργεί τους φακέλους που απαιτούνται και τα αρχεία που χρειάζονται ως βάσεις για να ξεκινήσει ο προγραμματιστής τη συγγραφή του κώδικα.

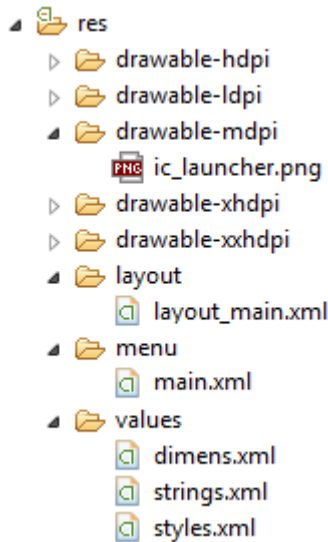
Οι φάκελοι που δημιουργούνται βοηθούν στην ταξινόμηση των αρχείων που προϋπάρχουν ή θα δημιουργηθούν αργότερα. Μέσα στον φάκελο src (source) βρίσκεται το αρχείο .java του πρώτου activity που εισήγαγε το Eclipse κατά τη δημιουργία του project (εικόνα 3.6). Σε αυτό το φάκελο εισάγονται αργότερα όλα τα αρχεία .java για την εφαρμογή.



Εικόνα 3.6: Ο φάκελος src και τα περιεχόμενά του.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, Package Explorer

Ένας άλλος σημαντικός φάκελος είναι ο res (resources). Μέσα σε αυτόν υπάρχουν αρκετοί υποφάκελοι όπως φαίνεται και στην εικόνα 3.7, όπου τοποθετούνται οι πόροι της εφαρμογής όπως τα γραφικά στοιχεία (φάκελοι drawable) όπως εικόνες και φόντο, τα αρχεία xml που περιέχουν τα layout της εφαρμογής (φάκελος layout), τα μενού (φάκελος menu) και το αρχείο strings (φάκελος values) που περιέχει τα κείμενα που χρησιμοποιεί η εφαρμογή. Ο προγραμματιστής μπορεί να προσθέσει περισσότερους φακέλους εάν το επιθυμεί.



Εικόνα 3.7: Φάκελος res και υποφάκελοι.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, Package Explorer

3.6 Ασφάλεια εφαρμογών Android

Όταν μια εφαρμογή Android εγκατασταθεί σε μια συσκευή κινητού, λειτουργεί με έναν δικό της μηχανισμό ασφαλείας (sandbox). Τα χαρακτηριστικά αυτού του μηχανισμού είναι τα εξής:

- Η πλατφόρμα του Android είναι ένα σύστημα Linux πολλαπλών χρηστών, στο οποίο η κάθε εφαρμογή θεωρείται ως ένας ξεχωριστός χρήστης.
- Η προεπιλογή του συστήματος είναι να απονέμει σε κάθε εφαρμογή μια μοναδική ταυτότητα (Linux user ID), η οποία χρησιμοποιείται και αναγνωρίζεται μόνο από το σύστημα και όχι από την εφαρμογή. Το σύστημα ορίζει τα δικαιώματα όλων των αρχείων μιας εφαρμογής έτσι ώστε να ταυτίζονται με το Linux user ID και να επιτρέπεται η πρόσβαση σε αυτά μέσω κατόπιν ταυτοποίησης.
- Κάθε διεργασία έχει τη δική της εικονική μηχανή, έτσι ώστε ο κώδικας μιας εφαρμογής να τρέχει ξεχωριστά από τις υπόλοιπες.
- Κάθε εφαρμογή τρέχει από προεπιλογή στη δική της διεργασία Linux. Το Android θέτει σε λειτουργία μια διεργασία όταν κάποιο από τα μέρη μιας εφαρμογής πρέπει να εκτελεστεί, και τις κλείνει όταν δεν είναι πια απαραίτητες ή όταν το σύστημα χρειάζεται ελεύθερη μνήμη για άλλες εφαρμογές.

Με αυτό τον τρόπο, το σύστημα αποκτάει το πλεονέκτημα της ασφαλείας καθώς μια εφαρμογή έχει πρόσβαση μόνο σε αυτά που χρειάζεται για να λειτουργήσει και τίποτα παραπάνω. Έτσι, μια εφαρμογή δεν έχει πρόσβαση σε μέρη του συστήματος για τα οποία δεν της έχει δοθεί η άδεια. Αυτό δεν αποκλείει το ενδεχόμενο για δυο εφαρμογές να μοιράζονται δεδομένα μεταξύ τους ή να έχουν πρόσβαση σε υπηρεσίες του συστήματος.

Υπάρχει η δυνατότητα να χρησιμοποιούν δυο εφαρμογές το ίδιο Linux user ID, πράγμα που σημαίνει ότι η μια μπορεί να έχει πρόσβαση στα αρχεία της άλλης. Για την συντήρηση των πόρων του συστήματος, εφαρμογές με το ίδιο Linux user ID μπορούν να τρέχουν και στην ίδια διεργασία και να μοιράζονται την εικονική μηχανή.

Επιπλέον, μια εφαρμογή μπορεί να ζητήσει άδεια πρόσβασης δεδομένων της συσκευής όπως οι επαφές του τηλεφωνικού καταλόγου του χρήστη, τα SMS, η εξωτερική κάρτα μνήμης και ουσιαστικά οπουδήποτε αλλού. Οι άδειες που απαιτούνται για μια τέτοια ενέργεια θα πρέπει να λαμβάνουν την έγκριση του χρήστη κατά την εγκατάσταση.

3.7 Συστατικά εφαρμογής

Τα συστατικά μιας εφαρμογής (App Components) απαρτίζουν τα δομικά στοιχεία της. Το κάθε component αλληλοεπιδρά με κάποιο τρόπο με το σύστημα ή συνεργάζεται με τα υπόλοιπα, επιτρέποντάς του την πρόσβαση στην εφαρμογή. Υπάρχουν τέσσερεις τύποι App Components, όπου ο κάθε ένας έχει δικό του κύκλο ζωής και έναν ξεχωριστό ρόλο σαν οντότητα.

- Δραστηριότητες (Activities). Σε μια εφαρμογή υπάρχουν συνήθως αρκετές διαφορετικές οθόνες ή σελίδες. Μια δραστηριότητα αντιπροσωπεύει μια οθόνη με ένα UI. Κάθε εφαρμογή λοιπόν, περιέχει τόσες δραστηριότητες, όσες και οι διαφορετικές οθόνες που περιέχει. Οι δραστηριότητες αυτές πρέπει να συνεργάζονται μεταξύ τους για να παράγουν ένα ολοκληρωμένο αποτέλεσμα. Για παράδειγμα, μια εφαρμογή μουσικής μπορεί να αποτελείται από μια δραστηριότητα που να δείχνει στο χρήστη τη λίστα των τραγουδιών, μια δραστηριότητα για την αναπαραγωγή ενός κομματιού και μια δραστηριότητα με πληροφορίες των τραγουδιών. Οι 3 αυτές

δραστηριότητες θα πρέπει να αλληλοεπιδρούν αρμονικά μεταξύ τους για να λειτουργήσει ορθά η μουσική εφαρμογή.

- Υπηρεσίες (Services). Μια υπηρεσία είναι μια λειτουργία η οποία τρέχει στο παρασκήνιο για την εκτέλεση ενεργειών μακροπρόθεσμα, ή για την εκτέλεση εργασιών για απομακρυσμένες διαδικασίες. Δεν παρέχουν κάποιο UI. Ένα παράδειγμα χρήσης τους είναι το εξής, μια υπηρεσία είναι αυτή που επιτρέπει σε κάποιον να ακούει μουσική με τη χρήση μιας εφαρμογής ενώ παράλληλα χρησιμοποιεί κάποια άλλη εφαρμογή. Μια δραστηριότητα μπορεί να θέσει σε εφαρμογή μια υπηρεσία ή να συνδεθεί με αυτήν με σκοπό την αλληλεπίδρασή τους.
- Πάροχοι Περιεχομένου (Content Providers). Ένας πάροχος περιεχομένου διαχειρίζεται ένα σύνολο πληροφοριών μιας εφαρμογής. Οι πληροφορίες αυτές μπορεί να είναι αποθηκευμένες στη συσκευή, σε μια βάση δεδομένων, στο ίντερνετ ή οπουδήποτε έχει πρόσβαση η εφαρμογή. Ο ρόλος του παρόχου περιεχομένου είναι να δίνει τη δυνατότητα σε άλλες εφαρμογές να έχουν πρόσβαση ή να μπορούν να τροποποιήσουν τα δεδομένα (εάν αυτό επιτρέπεται).
- Δέκτες Μετάδοσης (Broadcast Receivers). Ο δέκτης μετάδοσης είναι μια λειτουργία που είναι υπεύθυνη για μεταδόσεις ανακοινώσεων, από και προς το σύστημα και τις εφαρμογές. Μια ανακοίνωση μπορεί να είναι ένα σήμα, για παράδειγμα ότι ένα αρχείο έχει κατέβει και είναι διαθέσιμο προς χρήση. Παρόλο που αυτοί οι δέκτες δεν έχουν κάποιο δικό τους UI, μπορούν να ενημερώσουν το χρήστη για κάτι που έχει συμβεί δημιουργώντας ειδοποιήσεις στη μπάρα κατάστασης.

Οι δραστηριότητες, οι υπηρεσίες και οι δέκτες μετάδοσης ενεργοποιούνται από ένα ασύγχρονο μήνυμα που ονομάζεται Πρόθεση (Intent). Οι προθέσεις είναι αυτές που ενώνουν δυο ξεχωριστά components κατά την εκτέλεσή τους, ακόμη και αυτά που βρίσκονται σε ξεχωριστές εφαρμογές. Μια πρόθεση δημιουργείται με ένα αντικείμενο πρόθεσης (Intent Object), το οποίο ορίζει ένα μήνυμα ενεργοποίησης ενός συγκεκριμένου component ή ενός συγκεκριμένου τύπου component.

Για τις δραστηριότητες και τις υπηρεσίες, μια πρόθεση καθορίζει την ενέργεια που πρέπει να πραγματοποιηθεί. Για τους δέκτες μετάδοσης, η πρόθεση καθορίζει την ανακοίνωση προς μετάδοση. Οι πάροχοι περιεχομένου δεν ενεργοποιούνται από προθέσεις διότι ενεργοποιούνται από αιτήματα που λαμβάνουν από έναν Αναλυτή

Περιεχομένου (Content Resolver). Αυτός χειρίζεται όλες τις άμεσες συναλλαγές με τον πάροχο περιεχομένου, έτσι ώστε να μην απαιτείται επιπλέον χρήση κάποιου component πέραν μερικών μεθόδων.

3.8 Το αρχείο Manifest

Με τη δημιουργία νέου Project στο Eclipse, εμφανίζεται ένα αρχείο μέσα στο πακέτο της εφαρμογής που ονομάζεται Manifest.xml. Το αρχείο αυτό περιλαμβάνει στοιχεία της ονομασίας της εφαρμογής, όλα τα components της, ενώ όσα πρόκειται να δημιουργηθούν μελλοντικά θα πρέπει να τα εισάγει ο προγραμματιστής. Το Manifest περιέχει και είναι υπεύθυνο για:

- Την αναγνώριση των αδειών που ζητάει πρόσβαση η εφαρμογή (όλες οι άδειες που θα χρειαστεί η εφαρμογή για τη λειτουργία της εισάγονται στο Manifest).
- Τη δήλωση του ελάχιστου επιπέδου API που είναι απαραίτητο για τη λειτουργία της εφαρμογής.
- Τη δήλωση των χαρακτηριστικών λογισμικού ή υλικού της συσκευής που θα χρησιμοποιούνται ή θα απαιτούνται για τη λειτουργία της εφαρμογής.
- Τις βιβλιοθήκες API που μπορεί να είναι συνδεδεμένες με την εφαρμογή.

3.9 Σχεδιασμός της εφαρμογής

Για την υλοποίηση της εφαρμογής, το πρώτο βήμα ήταν ένας πρόχειρος σχεδιασμός του τι θα έπρεπε να περιέχει εμφανισιακά και τι ακριβώς καλούταν να χρησιμοποιήσει πρακτικά. Ο σχεδιασμός έγινε με γνώμονα την ευκολία χρήσης και την απλότητα, σε συνδυασμό με τη χρηστικότητα της εφαρμογής.

Όπως αναφέρθηκε και στους στόχους της εφαρμογής, από λειτουργικής άποψης η εφαρμογή θα χρησιμοποιεί το GPS για τη λήψη συντεταγμένων της τοποθεσίας που θα βρίσκεται ο εκάστοτε χρήστης της, καθώς και τη φωτογραφική μηχανή για λήψη μιας εικόνας. Αυτά τα δυο αποτελούν μέρη του hardware της συσκευής.

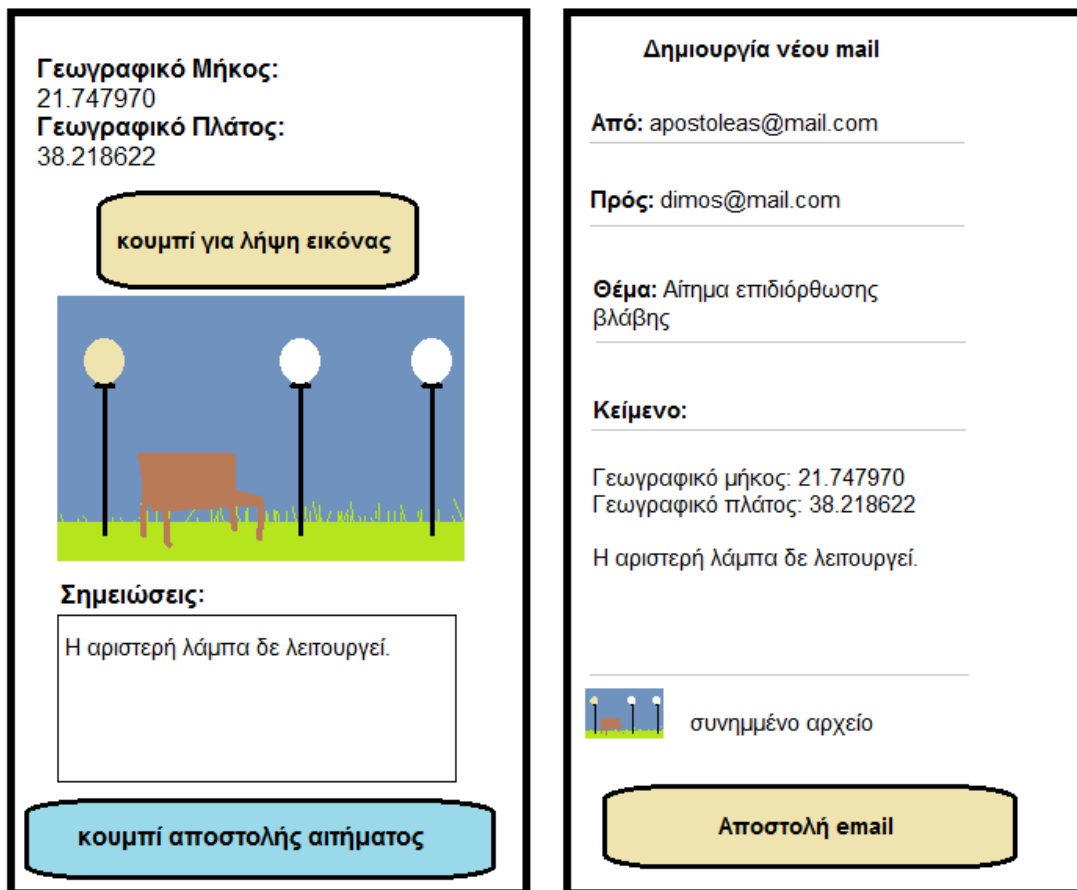
Επιπλέον, θα καλεί κατά την αποστολή ενός αιτήματος τις διαθέσιμες εφαρμογές email, δηλαδή κάποιο άλλο λογισμικό της συσκευής.

Συνοπτικά απαιτούνται:

- Μια αρχική οθόνη με επιλογές έναρξης της κύριας λειτουργίας και βοήθειας.

- Η οθόνη με την κύρια λειτουργία όπου θα περιέχονται:
 - Οι συντεταγμένες του GPS που θα λαμβάνονται αυτόματα.
 - Μια ειδοποίηση για την κατάσταση του GPS.
 - Μια εικόνα που θα λαμβάνεται εκείνη τη στιγμή και θα συνάπτεται στην οθόνη σε μικρογραφία.
 - Ένα πλαίσιο όπου θα μπορεί ο χρήστης να συμπληρώσει επιπλέον σημειώσεις.
 - Το κουμπί για την αποστολή του αιτήματος.
- Η οθόνη της εφαρμογής του email όπου θα επισυνάπτεται η φωτογραφία και θα εισάγονται αυτόματα οι συντεταγμένες που έχουν ληφθεί και οι σημειώσεις που έχει γράψει ο χρήστης. Η διεύθυνση αποστολής και το θέμα θα πρέπει να είναι ήδη συμπληρωμένα για άμεση αποστολή.
- Αναδυόμενα μενού για παραπομπή του χρήστη στις ρυθμίσεις GPS, τη βοήθεια και πληροφορίες σχετικά με την εφαρμογή.

Ο αρχικός σχεδιασμός του περιβάλλοντος της εφαρμογής στην εικόνα 3.8, δείχνει μια άποψη της τοποθέτησης των αντικειμένων στην οθόνη, καθώς και του πως θα φαίνεται το αποτέλεσμα στο email πριν την αποστολή του.



Οθόνη ενεργειών

Οθόνη αποστολής email

Εικόνα 3.8: Πρόχειρος σχεδιασμός της κύριας οθόνης και του email.

Πηγή: Σχεδιασμένη ειδικά για την εφαρμογή

Συγκεντρώνοντας τις παραπάνω απαιτήσεις, μπορούν να δημιουργηθούν οι κλάσεις, τα layout και τα μενού που θα χρησιμοποιηθούν για την υλοποίηση της εφαρμογής.

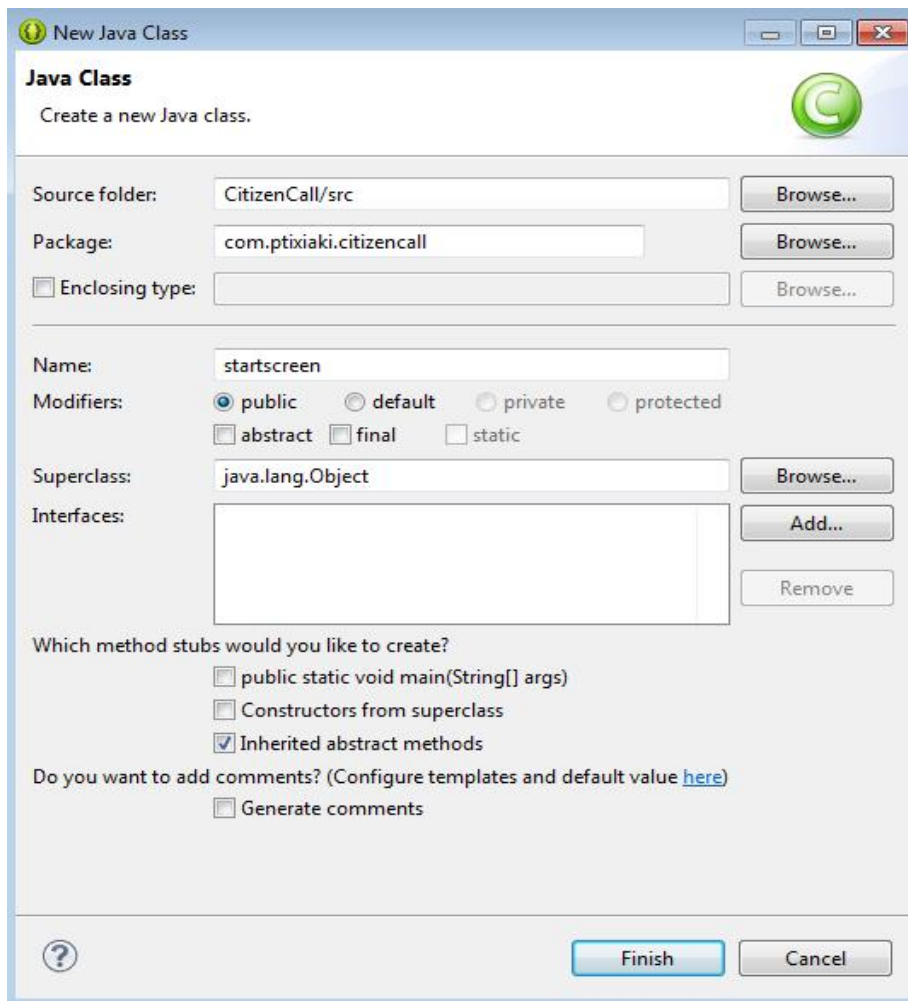
3.9.1 Καθορισμός των Δραστηριοτήτων (Activities)

Η εφαρμογή θα χρησιμοποιεί συνολικά τρεις οθόνες που θα καλύπτουν όλη την επιφάνεια της οθόνης του κινητού και μια μικρότερη με τις πληροφορίες της εφαρμογής. Η κάθε μια από αυτές τις τέσσερις οθόνες έχει τη δική της κλάση δραστηριότητας (activity class).

- Main, η δραστηριότητα της αρχικής οθόνης όταν ανοίγει η εφαρμογή. Περιέχει τα κουμπιά της Έναρξης και της Βοήθειας. Σε αυτήν την οθόνη βρίσκεται το αναδυόμενο μενού About με πληροφορίες σχετικά με την εφαρμογή.

- Startscreen, η οθόνη που εμφανίζεται στο χρήστη με το πάτημα του κουμπιού “Εναρξη” στην αρχική οθόνη. Εδώ λαμβάνουν μέρος οι κύριες λειτουργίες των αισθητήρων.
- Helpscreen, η οθόνη που εμφανίζεται στο χρήστη με το πάτημα του κουμπιού “Βοήθεια” στην αρχική οθόνη. Παρέχει μια επεξήγηση για το πώς λειτουργεί η εφαρμογή.
- About, στην ουσία ανοίγει ένα μικρότερο παράθυρο επάνω από την αρχική οθόνη με πληροφορίες σχετικά με την εφαρμογή.

Οι νέες δραστηριότητες, πλην της Main που αποκτήθηκε κατά τη δημιουργία του νέου Project, πρέπει να δημιουργηθούν χειροκίνητα. Έχουν θέση στον ίδιο φάκελο που βρίσκεται και η Main, τον src. Για τη δημιουργία τους στο Eclipse υπάρχουν δυο τρόποι, δεξί κλικ στον φάκελο του src – New – Class ή File – New – Class. Στην οθόνη που εμφανίζεται (εικόνα 3.9), αναγράφεται ο φάκελος στον οποίο θα τοποθετηθεί η νέα κλάση, όπως και το πακέτο στο οποίο ανήκει. Απαιτείται η συμπλήρωση του ονόματος της καινούριας κλάσης. ενώ παρέχονται και επιπλέον επιλογές για αυτόματη δημιουργία μεθόδων. Πατώντας το Finish δημιουργείται η καινούρια κλάση.



Εικόνα 3.9: Δημιουργία καινούριας κλάσης.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, New Class

3.9.2 Δήλωση των Δραστηριοτήτων στο Manifest

Για να λειτουργήσουν οι δραστηριότητες μιας εφαρμογής πρέπει να καταγράφονται στο Manifest.xml. Παρακάτω είναι η δήλωση της κλάσης Main στο αρχείο Manifest. Τοποθετείται εν μέσω των tags του application, το οποίο περιέχει την παράμετρο icon που ορίζει το εικονίδιο της εφαρμογής, την label που περιέχει την ονομασία της εφαρμογής και την theme για την εμφάνισή της.

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

Η Main ορίζεται ως activity. Έχει τεθεί ο προσανατολισμός της οθόνης ως κλειδωμένος σε πορτραίτο, ώστε να μην εναλλάσσεται οριζοντίως όταν το κινητό έχει κλίση. Μέσα στο intent-filter αναγράφεται τι θα συμβαίνει όταν καλείται η Main. Στην παράμετρο category, η κατάληξη Launcher ορίζει ότι αυτή θα είναι η πρώτη δραστηριότητα που θα ξεκινήσει όταν τεθεί σε λειτουργία η εφαρμογή. Υπάρχει μόνο μια δραστηριότητα δηλωμένη ως Launcher σε ένα Manifest, όλες οι υπόλοιπες δηλώνονται σαν Default.

```
<activity
    android:name="com.ptixiaki.citizencall.Main"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

Μέσα στο κυρίως σώμα του application θα συμπληρωθούν και οι υπόλοιπες δραστηριότητες. Για παράδειγμα, το startscreen.

```
<activity
    android:name="com.ptixiaki.citizencall.startscreen"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action
            android:name="com.ptixiaki.citizencall.STARTSCREEN"/>
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Στο category χρησιμοποιεί την κατάληξη Default. Αυτό χρησιμοποιείται γιατί η εν λόγω δραστηριότητα θα έρχεται στο προσκήνιο μόνο όταν καλείται.

Ομοίως με την startscreen, συμπληρώνεται παρακάτω και η helpscreen.

```
<activity
    android:name="com.ptixiaki.citizencall.helpscreen"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="com.ptixiaki.citizencall.HELPSCREEN"
        />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Η δραστηριότητα του about περιέχει διαφορετικό θέμα, για το λόγο ότι έχει σαν στόχο να εμφανίζεται σε ένα πλαίσιο το οποίο θα καταλαμβάνει όσο χώρο χρειάζεται για να προβάλλει τις πληροφορίες που περιέχει το layout του.

```
<activity
    android:name="com.ptixiaki.citizencall.about"
```

```

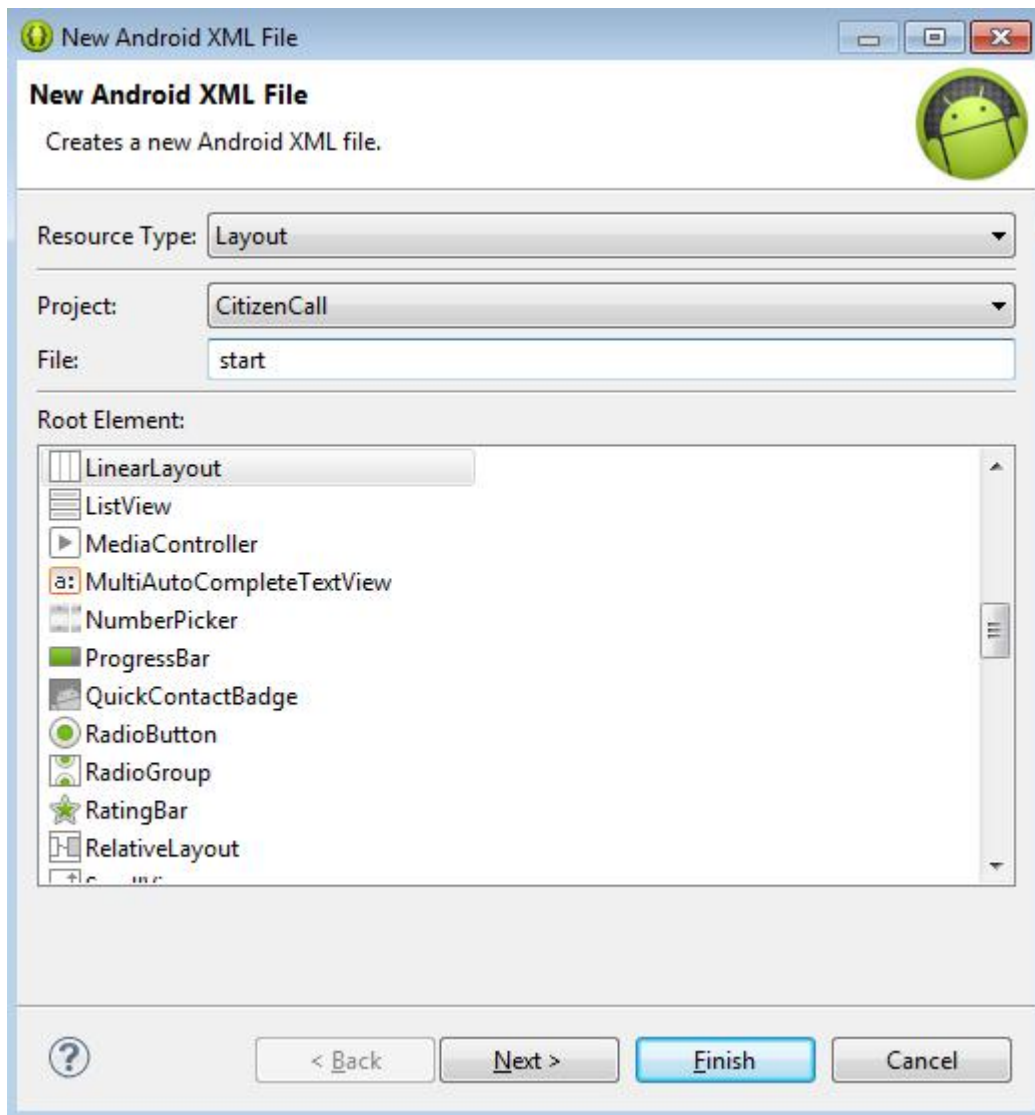
        android:label="@string/app_name"
        android:theme="@android:style/Theme.Dialog"
        android:screenOrientation="portrait" >
        <intent-filter>
            <action android:name="com.ptixiaki.citizencall.ABOUT" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>

```

3.9.3 Δημιουργία Layout

Μετά τη δήλωση των δραστηριοτήτων στο Manifest, σειρά έχει η δημιουργία ενός layout για το καθένα. Τα layout είναι γραμμένα σε γλώσσα xml, και τοποθετούνται στο Eclipse στον υποφάκελο του res – layout. Μέσα σε αυτά αναγράφονται οι πληροφορίες της εμφάνισης μιας δραστηριότητας και μπορεί ο προγραμματιστής να ορίσει τις ακριβείς τοποθεσίες των αντικειμένων που θέλει να εισάγει, όπως για παράδειγμα ένα κουμπί ή μια εικόνα ή το φόντο της δραστηριότητας.

Για τη δημιουργία νέων αρχείων layout, ακολουθείται το path File – New – Android XML File. Στο όνομα του αρχείου δεν επιτρέπονται κεφαλαία γράμματα ή σημεία στίξης πέραν της κάτω παύλας. Στην εικόνα 3.10 φαίνεται το παράθυρο που εμφανίζεται κατά τη δημιουργία νέου layout. Ο προγραμματιστής μπορεί να επιλέξει την τοποθεσία που θα αποθηκευτεί το αρχείο, το Project που θα ανήκει και τον τύπο που θα έχει.



Εικόνα 3.10: Δημιουργία νέου Layout.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, New Android XML File

Σε αυτή την εφαρμογή, όλες οι δραστηριότητες έχουν από ένα layout. Από τη δημιουργία του project υπάρχει διαθέσιμο το layout_main.xml, το οποίο ανήκει στη δραστηριότητα Main. Οπότε δημιουργούνται χειροκίνητα:

- start.xml – το layout της startscreen, σε μορφή RelativeLayout.
- help.xml – το layout της helpscreen, σε μορφή LinearLayout.
- about.xml – το layout του about, σε μορφή RelativeLayout.

Ένα αρχείο xml περιέχει πληροφορίες για την έκδοση και την κωδικοποίησή του.

Το RelativeLayout ορίζει τον τρόπο με τον οποίο θα εμφανίζονται και θα ευθυγραμμίζονται οι εντολές που αναγράφονται ανάμεσα στα tags του. Το

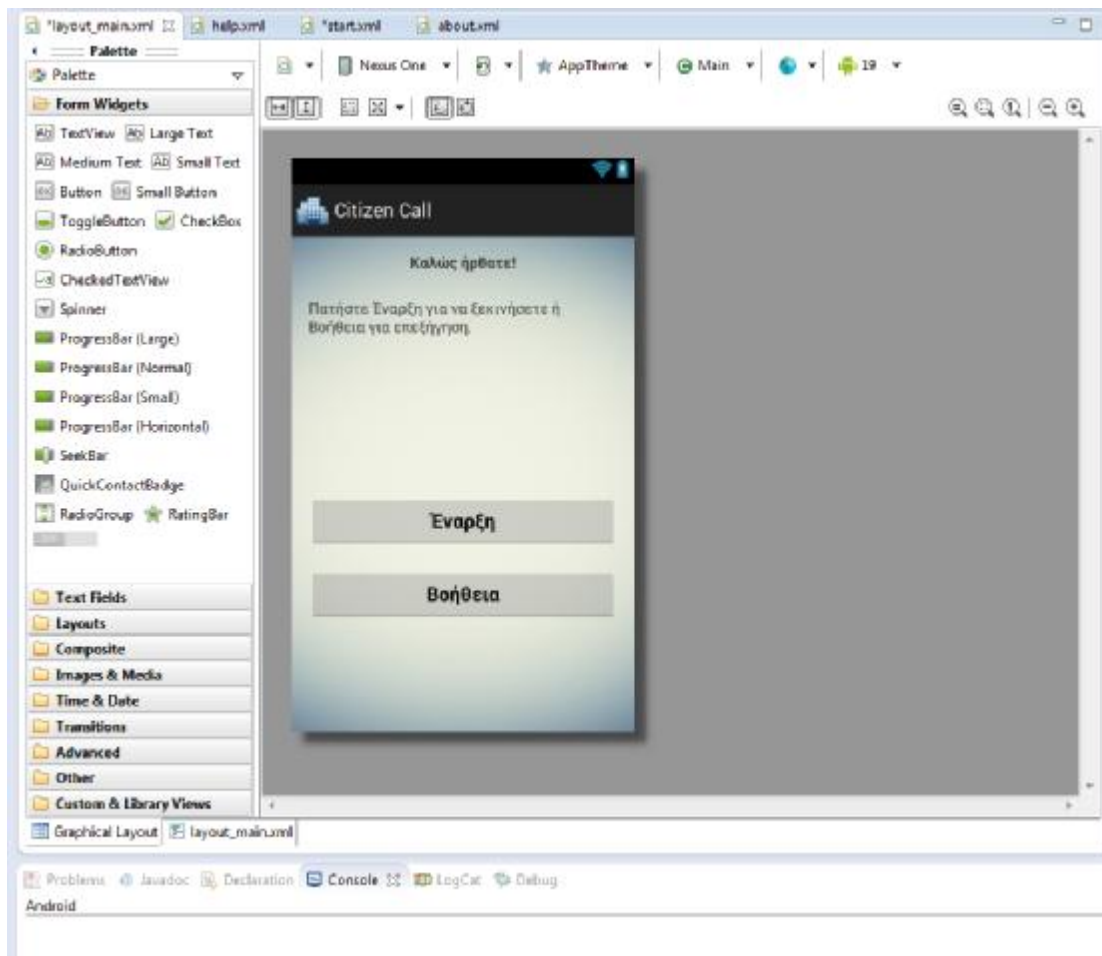
RelativeLayout δίνει την επιλογή να τοποθετούνται τα αντικείμενα οπουδήποτε στην οθόνη και να μπορούν να ευθυγραμμιστούν χρησιμοποιώντας σαν άξονες άλλα αντικείμενα. Μια εναλλακτική είναι το LinearLayout, με το οποίο οι εντολές εμφανίζονται με τη σειρά που αναγράφονται καθέτως.

Αρχικά ορίζεται η εμφάνιση του layout μέσω των layout_width και layout_height. Το match_parent δηλώνει ότι καλύπτει όλο το διαθέσιμο πλάτος και ύψος αντίστοιχα. Το φόντο του layout ορίζεται μέσω της εντολής background, όπου καλεί μια εικόνα με την ονομασία background από το φάκελο drawable.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >
```

Ο προγραμματιστής είναι υπεύθυνος για την εύρεση φόντου και την τοποθέτησή του σε έναν φάκελο drawable στο workspace. Επιπλέον, εντός των αγκυλών του RelativeLayout μπορούν να προστεθούν κι άλλες επιλογές που να αφορούν ολόκληρο το layout.

Το Eclipse δίνει στον προγραμματιστή τη δυνατότητα να δει το layout που έχει δημιουργήσει, χωρίς να χρειαστεί να ανοίξει έναν εξομοιωτή. Η επιλογή αυτή βρίσκεται στο κάτω μέρος της οθόνης και ονομάζεται Graphical Layout (εικόνα 3.11). Μπορεί να εναλλάσσεται μεταξύ του κώδικα xml και του Graphical Layout όσο συχνά θέλει για να ελέγξει αυτό που έχει φτιάξει.



Εικόνα 3.11: Το Graphical Layout του layout_main.xml.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, Graphical Layout

Μέσα στο Graphical Layout περιέχονται όλα τα αντικείμενα που μπορεί να εισάγει ο προγραμματιστής σε ένα layout, σε μορφή drag and drop. Η στοίχιση γίνεται απλούστερα μέσω του Graphical Layout, σέρνοντας το αντικείμενο στην επιθυμητή θέση. Το Eclipse παρέχει πληροφορίες ευθυγράμμισης για διευκόλυνση.

3.9.4 Δημιουργία Layout των Μενού

Τα Μενού αναδύονται στο κάτω μέρος της οθόνης, όταν ο χρήστης πατήσει το κουμπί του μενού στη συσκευή του. Μπορούν να περιέχουν οτιδήποτε, όπως ρυθμίσεις, επιλογές βοήθειας, έξοδο από την εφαρμογή ή το About, το οποίο συνήθως απεικονίζεται ως ένα παραθυράκι με πληροφορίες σχετικά με την εφαρμογή και τον προγραμματιστή. Στην εφαρμογή θα προστεθεί το About στην οθόνη της Main, και στο Μενού της startscreen θα προστεθούν οι Ρυθμίσεις του GPS και η Βοήθεια. Το layout του About έχει δημιουργηθεί στον φάκελο των

Μενού και ονομάζεται `aboutus.xml`, και το layout του Μενού της `startscreen` ονομάζεται `optionsmenu.xml`.

3.10 Προετοιμασία εικόνων και κειμένου

Το σύνολο των οπτικοακουστικών στοιχείων, καθώς και των αλφαριθμητικών κειμένων, τοποθετούνται στον φάκελο `res`. Σε αυτή την εφαρμογή θα εισαχθούν στοιχεία στο αρχείο `strings.xml` και στους φακέλους `drawable`.

3.10.1 Η χρήση των String στο Android

Στον φάκελο `res – values` βρίσκεται το αρχείο `strings.xml`. Το αρχείο αυτό περιλαμβάνει μια σειρά μεταβλητών που καλούνται σε διαφορετικά σημεία της εφαρμογής και περιέχουν κείμενο ή και αριθμούς. Για την αρχική οθόνη, έχουν εισαχθεί το κείμενο για το κουμπί της Έναρξης και της Βοήθειας. Το string του ονόματος της εφαρμογής εμφανίζεται αυτόματα κατά την δημιουργία νέου Project.

```
<resources>
```

```
    <string name="app_name">Citizen Call</string>
    <string name="hello">Καλώς ήρθατε!</string>
    <string name="start">Έναρξη</string>
    <string name="help">Βοήθεια</string>
```

```
</resources>
```

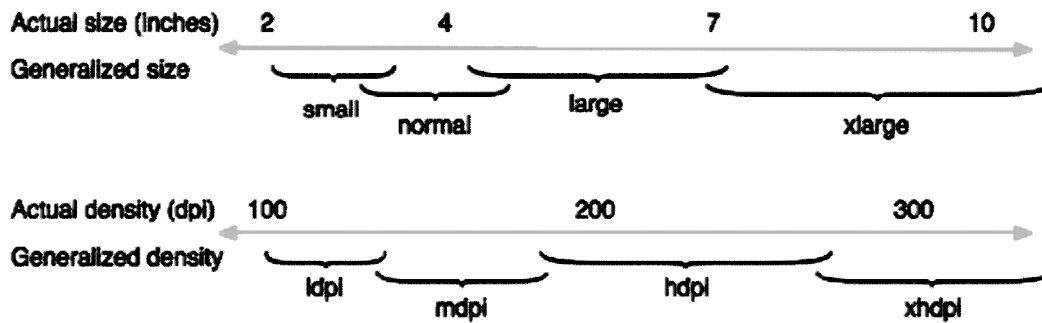
Εντός του `resources` τοποθετούνται όλα τα `string`. Το κάθε `string` έχει μια μοναδική ονομασία που λειτουργεί ως ταυτότητά του. Στη συνέχεια της εφαρμογής θα προστεθούν περισσότερα `string` για όλα τα πεδία που χρησιμοποιούν κάποιο κείμενο ως τίτλο ή περιεχόμενο.

3.10.2 Οι φάκελοι γραφικών Drawable

Μέσα στον φάκελο `res` βρίσκεται ένα σύνολο φακέλων με κοινή ονομασία, `drawable`, και διαφορετική κατάληξη. Οι φάκελοι αυτοί θα φιλοξενήσουν τα γραφικά στοιχεία της εφαρμογής, καταναμεμένα αναλόγως με το μέγεθος της ανάλυσης της οθόνης που απευθύνονται. Στο όνομα των φακέλων ορίζεται και το μέγεθος το οποίο φιλοξενεί ο κάθε φάκελος. Το Android επιλέγει αυτόματα την κατάλληλη ανάλυση μιας εικόνας από τον σωστό φάκελο, ανάλογα με το μέγεθος της οθόνης της συσκευής που χρησιμοποιείται (εικόνα 3.12). Οι κατηγορίες των φακέλων βάσει της ανάλυσης της οθόνης είναι:

- `ldpi` – low density (χαμηλή ανάλυση)

- mdpi – medium density (μέτρια ανάλυση)
- hdpi – high density (υψηλή ανάλυση)
- xhdpi – extra high density (πολύ υψηλή ανάλυση)
- xxhdpi – extra extra high density (πάρα πολύ υψηλή ανάλυση)



Εικόνα 3.12: Διαφορετικά μεγέθη οθόνης σε σχέση με τα dpi.

Πηγή: <http://developer.android.com>, Άρθρο *Supporting Multiple Screens*

Για την εφαρμογή θα εισαχθούν στους φακέλους drawable εικονίδια για τα κουμπιά των Μενού, μια εικόνα φόντου, το εικονίδιο που χρησιμοποιεί η εφαρμογή, και λοιπές βοηθητικές εικόνες.

ΚΕΦΑΛΑΙΟ 4

Υλοποίηση Εφαρμογής

4.1 Εισαγωγή

Το στάδιο που ακολουθεί μετά τον σχεδιασμό και την ανάλυση των περιεχομένων που θα έχει μια εφαρμογή, είναι η υλοποίηση αυτής. Αυτό απαιτεί ανάπτυξη κώδικα java και xml, δοκιμαστικές εκδόσεις και επιδιόρθωση των σφαλμάτων που μπορεί να εμφανιστούν. Είναι το πιο χρονοβόρο και σημαντικό στάδιο για την δημιουργία μιας εφαρμογής.

4.2 Υλοποίηση της Αρχικής Οθόνης

Η Αρχική Οθόνη είναι το πρώτο πράγμα που θα βλέπει ο χρήστης της εφαρμογής όταν εισέρχεται σε αυτήν. Βάσει του σχεδιασμού, θα περιέχει δυο κουμπιά, ένα της Οθόνης Έναρξης και ένα της Οθόνης Βοήθειας. Επίσης, θα περιέχει ένα αναδυόμενο Μενού που θα περιέχει το About.

4.2.1 Το layout_main.xml

Το layout start έχει δημιουργηθεί χρησιμοποιώντας RelativeLayout. Οι παράμετροι που επηρεάζουν όλο το layout αναγράφονται εντός του πεδίου του RelativeLayout.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/background"
android:orientation="vertical"
android:paddingBottom="@dimen/acti_vity_vertical_margin"
android:paddingLeft="@dimen/acti_vity_hori_zontal_margin"
android:paddingRight="@dimen/acti_vity_hori_zontal_margin"
android:paddingTop="@dimen/acti_vity_vertical_margin"
tools:context=". Main" >
```

Το padding σχηματίζει ένα αόρατο πλέγμα στις άκρες της οθόνης, ώστε να μην ακουμπάνε τα αντικείμενα στις άκρες της. Έχει οριστεί επίσης το φόντο που θα χρησιμοποιηθεί για όλες τις οθόνες της εφαρμογής.

Στο πάνω μέρος της οθόνης βρίσκονται δυο ξεχωριστά κείμενα καλωσορίσματος.

Το κείμενο χρησιμοποιεί το TextView για να προβληθεί.

```
<TextView
android:id="@+id/hello"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:text="@string/hello"
android:textStyle="bold" />
```

Στο παραπάνω TextView χρησιμοποιούνται τέσσερις απαραίτητες για την ύπαρξη του παράμετροι.

- Id: πρόκειται για την ταυτότητα ενός αντικειμένου. Είναι ξεχωριστή για κάθε αντικείμενο και εισάγεται με την εντολή “@+id/” και μια λέξη.
- Layout_width: το πλάτος που επιτρέπεται να καταλαμβάνει το κείμενο. Το wrap_content του επιτρέπει να καταλάβει όσο χώρο χρειάζεται για να φαίνεται ολόκληρο.
- Layout_height: το ύψος που επιτρέπεται να έχει το κείμενο. Με το wrap_content συμβαίνει το ίδιο με το πλάτος, μπορεί να έχει όσο ύψος χρειάζεται.
- Text: το κείμενο που θα εμφανίζεται. Ορίζεται με τη χρήση του ανάλογου string.

Η παράμετρος layout_centerHorizontal ορίζει την στοίχιση του κειμένου, στην προκειμένη περίπτωση βρίσκεται στο κέντρο. Το textStyle παρέχει επιλογές κειμένου όπως έντονη γραφή (bold), πλάγια (italic) ή χρήση και των δυο.

Όταν ένα καινούριο αντικείμενο προστίθεται στην οθόνη μέσω του Graphical Layout, εμφανίζεται αυτόματα στο xml και χρειάζεται ορισμό. Φυσικά, μπορεί να οριστεί εξ’ αρχής γραπτώς στο xml. Το κουμπί της Έναρξης, για παράδειγμα ορίζεται ως εξής:

<Button

```
android:id="@+id/Start"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_centerHorizontal="true"
android:layout_marginBottom="155dp"
android:text="@string/start"
android:textStyle="bold" />
```

Οι παράμετροι που θα πρέπει να περιέχει αναγκαστικά ένα κουμπί είναι:

- Id
- Layout_width
- Layout_height
- Text

Οι εντολές `layout_alignParentBottom`, `layout_centerHorizontal` και `layout_marginBottom` απευθύνονται στην ευθυγράμμιση του κουμπιού ως προς το χώρο. Εάν ο προγραμματιστής αλλάξει την τοποθεσία του κουμπιού στο Graphical Layout κουνώντας το με το ποντίκι, εμφανίζονται αυτόματα οι εντολές της τοποθεσίας στο xml, στις παραμέτρους του αντίστοιχου κουμπιού. Ομοίως με το κουμπί της Έναρξης, προστίθεται και ορίζεται και το κουμπί της Βοήθειας.

```
<Button
    android:id="@+id/Help"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Start"
    android:layout_alignTop="@+id/Start"
    android:layout_marginTop="68dp"
    android:text="@string/help"
    android:textStyle="bold" />
```

4.2.2 Η κλάση Main

Για την κλάση Main έχει δημιουργηθεί το layout που θα χρησιμοποιεί. Μέσα σε αυτό υπάρχουν δυο κουμπιά που θα πρέπει να αποκτήσουν λειτουργικότητα, αλλά για αρχή θα πρέπει να οριστεί το `layout_main.xml` ως layout της κλάσης Main.

```
package com.ptixiakı.cıti zencall;
public class Main extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_main);
    }
```

Εντός της δραστηριότητας `onCreate` προστίθεται η μέθοδος `setContentView(R.layout.layout_main)`, η οποία ορίζει το `layout_main` στην αντίστοιχη κλάση. Το `R` χρησιμοποιείται για να κληθούν πόροι και ακολουθείται από το φάκελο που βρίσκεται το περιεχόμενο που θα χρησιμοποιηθεί, και την ονομασία αυτού.

Για τη λειτουργία ενός κουμπιού θα πρέπει πρώτα να οριστεί το κουμπί στην δραστηριότητα. Για παράδειγμα, το κουμπί της Έναρξης:

```
Button bStart = (Button) findViewById(R.id.Start);
```

Το `Button` θα χρειαστεί `Import` για να χρησιμοποιηθεί από την κλάση. Το Eclipse το εμφανίζει αρχικά υπογραμμισμένο, και με την τοποθεσία του ποντικιού επάνω

του, εμφανίζεται η επιλογή Import. Το κουμπί ταυτίζεται με αυτό στο layout της κλάσης βρίσκοντάς το μέσω της ταυτότητάς του.

Η λειτουργία του κουμπιού ορίζεται ξεχωριστά.

```
bStart.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        startActivity(new Intent("com.ptixiaki.citizencall.STARTSCREEN"));  
    }  
});
```

Παραπάνω φαίνεται ότι η μεταβλητή που χρησιμοποιείται για το κουμπί της Έναρξης, καλεί μια άλλη δραστηριότητα οθόνης, την startscreen.

Με τον ίδιο τρόπο ορίζεται και το κουμπί της Βοήθειας.

```
bHelp.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        startActivity(new Intent("com.ptixiaki.citizencall.HELPSCREEN"));  
    }  
});
```

4.2.3 Το Μενού About

Το Μενού που θα αναδύεται στην Αρχική Οθόνη θα περιέχει μια μόνο επιλογή, το About. Μέσα στο αρχείο layout aboutus στον φάκελο menu, τοποθετούνται οι πληροφορίες του κουμπιού.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >  
  
<item  
    android:id="@+id/aboutus"  
    android:title="@string/about"  
    android:icon="@drawable/ic_info"/>  
  
</menu>
```

Το item περιέχει μια ταυτότητα, το κείμενο του τίτλου του και μια εικόνα.

Το περιεχόμενο της οθόνης που θα εμφανίζεται όταν πατιέται το κουμπί, περιέχεται στο layout about.xml.

Χρειάζεται όμως και μια δραστηριότητα η οποία θα καλεί αυτό το layout, η οποία βρίσκεται στην κλάση about.java.

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

```

        setContentView(R.layout. about);
    }
}

```

Η onCreate θέτει το layout και είναι έτοιμη προς χρήση.

Μέσα στην Main.java περιλαμβάνεται ο κώδικας που καλεί το αναδυόμενο Μενού.

```

@Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu. aboutus, menu);
        return true;
    }

```

Σε αυτό το σημείο καλεί το layout του Μενού aboutus που βρίσκεται στον αντίστοιχο φάκελο.

```

    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()) {
            case R.id. aboutus:
                Intent a = new Intent("com.ptixiaki.citizencall.ABOUT");
                startActivity(a);
            }
        return false;
    }

```

Όταν επιλέγεται το κουμπί του About, ανοίγει η δραστηριότητά του που περιλαμβάνει ένα layout με πληροφορίες. Το τελικό αποτέλεσμα της Αρχικής Οθόνης με το αναδυόμενο Μενού φαίνεται στην εικόνα 4.1.



Εικόνα 4.1: Η Αρχική Οθόνη με το αναδυόμενο Μενού.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του AVD Simulator μέσω του Eclipse

4.3 Υλοποίηση της Οθόνης Έναρξης

Είναι η πιο σημαντική οθόνη καθώς εδώ λαμβάνουν χώρα οι περισσότερες λειτουργίες της εφαρμογής. Σε αυτή την οθόνη καλούνται οι αισθητήρες του GPS και της φωτογραφικής μηχανής και επιστρέφουν τα αντίστοιχα αποτελέσματα, προσθέτει ο χρήστης σημειώσεις και γίνεται η αποστολή του αιτήματος. Αρχικά, θα πρέπει να υλοποιηθεί το layout της οθόνης αυτής, το start.xml.

4.3.1 Το layout start.xml

Το layout αυτό θα χρησιμοποιεί TextView για τα κείμενα, Button για τα δυο κουμπιά, ένα ImageView για να απεικονίζεται μια μικρογραφία της εικόνας που θα λαμβάνεται και ένα EditText που θα δέχεται τις σημειώσεις.

Οι συντεταγμένες του GPS θα εμφανίζονται κάτω από τα πεδία με id Longitude και Latitude, στα showLongitude και showLatitude αντίστοιχα.

```
<TextView
    android:id="@+id/Longitude"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/hello"
    android:layout_marginTop="22dp"
    android:text="@string/longitude"
    android:textStyle="bold" />

<TextView
    android:id="@+id/showLongitude"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text=""
    android:layout_below="@+id/Longitude"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/Latitude"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="@string/latitude"
    android:layout_below="@+id/showLongitude"
    android:textStyle="bold"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/showLatitude"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text=""
    android:layout_below="@+id/Latitude"
    android:layout_alignParentLeft="true"/>
```

Τα πεδία που θα περιέχουν τις συντεταγμένες, έχουν κενή την παράμετρο του κειμένου ώστε να εισαχθούν αργότερα οι συντεταγμένες που θα ληφθούν.

Για τη λήψη της εικόνας και την εμφάνισή της χρησιμοποιείται ένα Button και από κάτω του θα βρίσκεται το ImageView.

```
<Button
    android:id="@+id/camera"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/showLatitude"
    android:layout_centerHorizontal="true"
    android:text="@string/camera_photo" />
```

<ImageView

```
android:id="@+id/camera_photo"
android:layout_width="160dp"
android:layout_height="100dp"
android:layout_below="@+id/camera"
android:layout_centerHorizontal="true"
android:contentDescription="@string/photo_desc" />
```

Οι διαστάσεις της μικρογραφίας ορίζονται σε dp (Density-independent Pixels), μια μονάδα που δεν μεταφράζεται σε ακριβές μέγεθος καθώς εναλλάσσεται ανάλογα με το μέγεθος της οθόνης της συσκευής που χρησιμοποιεί την εφαρμογή.

Οι Σημειώσεις θα τοποθετηθούν κάτω από το ImageView, και το αντίστοιχο TextView που δηλώνει την ονομασία του πεδίου.

<TextView

```
android:id="@+id/notes_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/camera_photo"
android:layout_centerHorizontal="true"
android:layout_marginTop="15dp"
android:text="@string/notes"
android:textStyle="bold" />
```

<EditText

```
android:id="@+id/notes"
android:layout_width="wrap_content"
android:layout_height="50dp"
android:layout_above="@+id/send"
android:layout_below="@+id/notes_text"
android:layout_centerHorizontal="true"
android:background="#ffffff"
android:ems="10"
android:inputType="textMultiLine" >
```

```
<requestFocus />
```

```
</EditText>
```

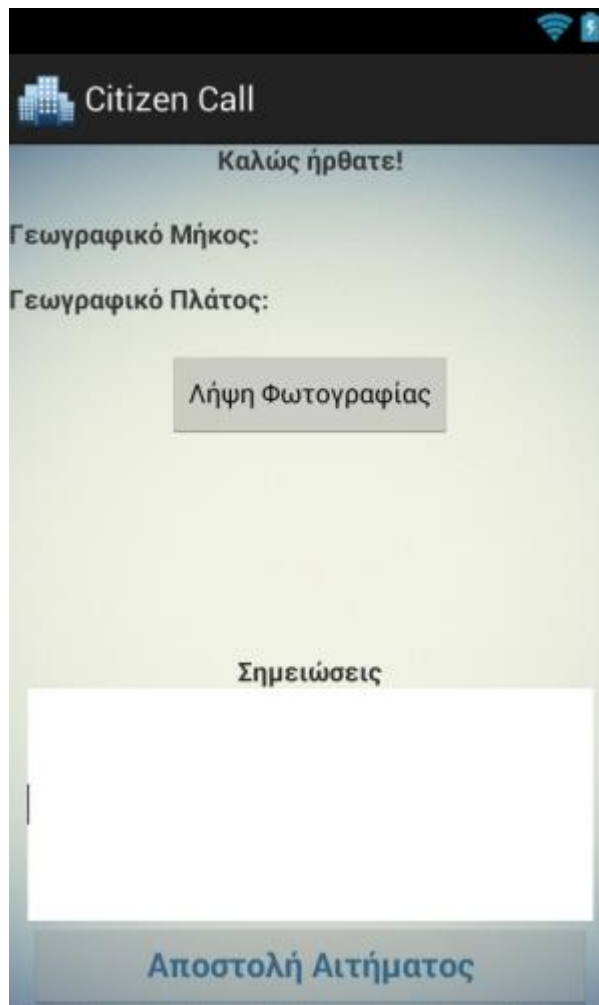
Το requestfocus είναι η εντολή που εμφανίζει το πληκτρολόγιο, όταν αυτό είναι απαραίτητο για την συμπλήρωση κειμένου. Βρίσκεται εντός του EditText, δηλαδή του πεδίου των σημειώσεων.

Τέλος, το κουμπί της αποστολής θα εμφανίζεται στο κάτω μέρος της οθόνης.

<Button

```
android:id="@+id/send"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:text="@string/send"
android:textStyle="bold"
android:textColor="#336688" />
```

Το συνολικό αποτέλεσμα του layout της οθόνης Έναρξης φαίνεται στην εικόνα 4.2.



Εικόνα 4.2: Το layout της οθόνης Έναρξης.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, Graphical Layout

4.3.2 Η κλάση `startscreen.java`

Σε αυτή την κλάση βρίσκεται το μεγαλύτερο κομμάτι του κώδικα της εφαρμογής. Γίνεται χρήση των κατάλληλων υπηρεσιών για τη λειτουργία του GPS και της φωτογραφικής μηχανής, και ορίζονται τα κουμπιά.

Στην κλάση αυτή πρέπει να αρχικά οριστεί το layout που θα χρησιμοποιηθεί. Με τον ίδιο τρόπο που ορίστηκε και στην Main, ορίζεται και εδώ.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.start);
}
```

Το GPS πρέπει να λειτουργήσει και να λάβει συντεταγμένες της τρέχουσας τοποθεσίας, τις οποίες θα εμφανίσει στην οθόνη στα πεδία του `showLongitude` και

showLatitude. Για την χρήση των υπηρεσιών εύρεσης τοποθεσίας ακολουθείται η κάτωθι διαδικασία, εντός της onCreate που δημιουργήθηκε παραπάνω.

```
final LocationManager lm =  
(LocationManager) getSystemService(Context.LOCATION_SERVICE);  
final LocationListener ll = new myLocationListener();  
lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, ll);
```

Το LocationManager παρέχει πρόσβαση στις υπηρεσίες εντοπισμού του συστήματος. Για τη χρήση του, χρησιμοποιείται το Context.getSystemService(Context.LOCATION_SERVICE).

Το LocationListener χρησιμοποιείται για τη λήψη ειδοποιήσεων από το LocationManager, όταν η τοποθεσία του χρήστη έχει αλλάξει. Οι μέθοδοι αυτοί καλούνται εάν το LocationListener έχει καταχωρηθεί με την υπηρεσία διαχείρισης εντοπισμού θέσης χρησιμοποιώντας τη μέθοδο requestLocationUpdates(String, long, float, LocationListener).

Για τη χρήση των υπηρεσιών εύρεσης τοποθεσίας θα δημιουργηθεί μια νέα κλάση.

```
class myLocationListener implements LocationListener{
```

Μέσα σε αυτήν, γίνεται η λήψη των δεδομένων τοποθεσίας, η αποστολή αυτών προς εμφάνιση στα αντίστοιχα πεδία του layout, ενώ έχουν δημιουργηθεί και ειδοποιήσεις προς το χρήστη για την κατάσταση του GPS του.

Αναλυτικά, αρχικά δέχεται αποτελέσματα από το GPS όταν αλλάζει η τοποθεσία, και τα εισάγει στα πεδία που έχουν δημιουργηθεί στο layout με τη χρήση των μεταβλητών pLong και pLat.

```
@Override  
public void onLocationChanged(Location location) {  
    if(location != null)  
    {  
        double pLong = location.getLongitude();  
        double pLat = location.getLatitude();  
  
        latitude.setText(Double.toString(pLat));  
        longitude.setText(Double.toString(pLong));  
    }  
}
```

Επιπρόσθετα, εάν το GPS είναι απενεργοποιημένο στη συσκευή, ειδοποιεί τον χρήστη με μία σύντομη ειδοποίηση στην οθόνη για να το ενεργοποιήσει. Όταν αυτό ενεργοποιηθεί, εμφανίζεται μια ειδοποίηση επιβεβαίωσης και προτροπής προς το χρήστη να περιμένει μέχρις ότου εμφανιστούν στην οθόνη οι συντεταγμένες, καθώς μπορεί να πάρει λίγη ώρα.

```

@Override
public void onProviderDisabled(String provider) {
    Toast.makeText(getApplicationContext(),
        getResources().getString(R.string.gps_off), Toast.LENGTH_SHORT).show();
}

@Override
public void onProviderEnabled(String provider) {
    Toast.makeText(getApplicationContext(),
        getResources().getString(R.string.gps_on), Toast.LENGTH_LONG).show();
}

@Override
public void onStatusChanged(String provider, int status,
    Bundle extras) {
    // TODO Auto-generated method stub
}
}

```

Για τη φωτογραφική μηχανή θα χρησιμοποιηθούν δυο νέες μεταβλητές, μια για την λήψη της εικόνας και μια για τη χρήση της ληφθείσας εικόνας.

```

private static int TAKE_PICTURE = 1;
private Uri imageUri;

```

Το κουμπί που έχει τοποθετηθεί στο layout για τη Λήψη Εικόνας, ορίζεται έτσι ώστε όταν το πατάει ο χρήστης να τίθεται σε λειτουργία η φωτογραφική μηχανή.

Για να συμβεί αυτό, γίνεται χρήση ενός Listener.

```

Button cameraButton = (Button) findViewById(R.id.camera);
cameraButton.setOnClickListener(cameraListener);
}

private OnClickListener cameraListener = new OnClickListener() {
    public void onClick(View v) {
        takePhoto(v);
    }
};

```

Το cameraListener χρησιμοποιείται έτσι ώστε να γίνει η μετάβαση από την οθόνη της Έναρξης στην οθόνη της φωτογραφικής μηχανής του κινητού για τη λήψη μιας εικόνας.

Τα δεδομένα της εικόνας που θα ληφθεί θα αποθηκευτούν σε ένα προσωρινό αρχείο που έχει δημιουργηθεί στην εξωτερική μνήμη του κινητού, και έχει ονομαστεί pic.jpg.

```

public void takePhoto(View v) {
    Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");
    File photo = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_
PICTURES), "pic.jpg");
    intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(photo));
    imageUri = Uri.fromFile(photo);
    startActivityForResult(intent, TAKE_PICTURE);
}

```

}

Αφού έχει αποφασιστεί το τι θα απογίνει η φωτογραφία αφού ληφθεί, έχει έρθει η ώρα να γίνει και η λήψη της. Η φωτογραφική μηχανή έρχεται σε πρώτο πλάνο και ο χρήστης εστιάζει στο πρόβλημα που έχει εντοπίσει και λαμβάνει μια φωτογραφία του.

Τα βήματα που ακολουθούνται παρακάτω είναι τα εξής:

1. Δημιουργία του onActivityResult για την απόκτηση αποτελεσμάτων.
2. Έλεγχος του requestCode. Εάν είναι 1 όπως έχει οριστεί νωρίτερα, τότε συμβαίνουν τα ακόλουθα.

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
```

```
    super.onActivityResult(requestCode, resultCode, data);  
    switch(requestCode) {
```

3. Εάν ο χρήστης έχει λάβει μια φωτογραφία, αυτή παίρνει τη θέση της μεταβλητής imageUrl που ορίστηκε παραπάνω.
4. Ενημέρωση των εφαρμογών για τις αλλαγές που έχουν συμβεί.
5. Αποκτάται πρόσβαση στο ImageView του start layout.

```
    case 1:
```

```
        if(resultCode == Activity.RESULT_OK) {  
            Uri selectedImage = imageUrl;  
            getContentResolver().notifyChange(selectedImage, null);  
            ImageView imageView =
```

```
(ImageView) findViewById(R.id.camera_photo);
```

6. Δημιουργείται ένα αντικείμενο Αναλυτή Περιεχομένου το οποίο θα μας επιτρέψει την πρόσβαση στο αρχείο της εικόνας που λήφθηκε.

```
        try {  
            BitmapFactory.Options options = new BitmapFactory.Options();  
            options.inSampleSize = 4;  
            AssetFileDescriptor fileDescriptor = null;  
            fileDescriptor =
```

```
getContentResolver().openAssetFileDescriptor(selectedImage, "r");
```

7. Λήψη του bitmap από το αρχείο εικόνας, με τη χρήση του Αναλυτή Περιεχομένου.
8. Τοποθέτηση του bitmap στο ImageView.

```
            Bitmap bitmap =  
            BitmapFactory.decodeFileDescriptor(fileDescriptor.getFileDescriptor(),  
            null, options);
```

```
            imageView.setImageBitmap(bitmap);  
            Toast.makeText(startscreen.this, selectedImage.toString(),  
            Toast.LENGTH_LONG).show();
```

9. Σε περίπτωση σφάλματος, ειδοποιείται ο χρήστης και καταγράφεται το συμβάν στο logtag.

```
    } catch(Exception e) {  
        Toast.makeText(startscreen.this, "failed to load",  
Toast.LENGTH_LONG).show();  
        Log.e(logtag, e.toString());  
    }  
}
```

Αφού γίνει η συλλογή στοιχείων από τους αισθητήρες του GPS και της φωτογραφικής μηχανής, πρέπει να γίνει η αποστολή αυτών, συμπεριλαμβανομένων των σημειώσεων. Οι μεταβλητές που θα χρειαστούν είναι οι:

```
TextView latitude;  
TextView longitude;  
EditText Notes;  
String message, sLong, sLat;  
String body = "";
```

Οι μεταβλητές αυτές χρειάζονται για την επικόλληση των στοιχείων που έχουν συλλεγεί στο κύριο μέρος του email. Αρχικά πρέπει να γίνει η ταυτοποίηση αυτών των μεταβλητών με τα αντίστοιχα πεδία που στοχεύουν η κάθε μια.

```
latitude = (TextView) findViewById(R.id.showLatitude);  
longitude = (TextView) findViewById(R.id.showLongitude);  
Notes = (EditText) findViewById(R.id.notes);
```

Έπειτα, ακολουθεί ο κώδικας του κουμπιού Αποστολής Αιτήματος. Το κουμπί τίθεται εκτός λειτουργίας αρχικά.

```
Button bSend = (Button) findViewById(R.id.send);  
bSend.setEnabled(false);  
bSend.setOnClickListener(new View.OnClickListener() {
```

```
    @Override  
    public void onClick(View v) {
```

Εντός των αγκυλών, γίνεται η μετατροπή των στοιχείων που έχουν ληφθεί από τις παραπάνω μεταβλητές σε απλό κείμενο και αριθμούς, για χρήση εντός του κυρίου μέρους του email.

Το κείμενο αυτό, συγκεντρώνεται στη μεταβλητή body.

```
message = Notes.getText().toString();  
sLong = longitude.getText().toString();  
sLat = latitude.getText().toString();  
body += "Longitude: " + sLong + "\n" + "Latitude: " + sLat + "\n" +  
message;
```

Έπειτα, ακολουθεί ο κώδικας που απαιτείται για την αποστολή ενός email, με τη συμπλήρωση όλων των πεδίων ώστε να είναι έτοιμο προς αποστολή κατά την εμφάνισή του.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);

    emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[] {"doki.miptixiakis@gmail.com"});
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Epidi orthosi
Provlimatos");
    emailIntent.putExtra(Intent.EXTRA_TEXT, body);
    emailIntent.setType("message/rfc822");
    emailIntent.putExtra(Intent.EXTRA_STREAM, imageUri);
    startActivity(Intent.createChooser(emailIntent, "Choose
an Email client"));
    }
});
```

Για τη δοκιμασία έχει οριστεί ένα πρόχειρο email λήψης των αιτημάτων. Κατά το πάτημα του κουμπιού της Αποστολής Αιτήματος, ο χρήστης καλείται να επιλέξει την εφαρμογή email που επιθυμεί να χρησιμοποιήσει.

Επειδή είναι απαραίτητη η ύπαρξη της εικόνας για την αποστολή του email, έχει τεθεί το κουμπί ως ανενεργό μέχρις ότου να ληφθεί μια. Στον κώδικα της εικόνας, αφότου γίνει η λήψη, συμπληρώνεται το κομμάτι που ενεργοποιεί το κουμπί της Αποστολής Αιτήματος.

```
Button bSend = (Button) findViewById(R.id.send);
bSend.setEnabled(true);
```

4.3.3 Δήλωση αδειών στο Manifest

Η εφαρμογή χρησιμοποιεί μια σειρά υπηρεσιών που απαιτούν ειδική άδεια. Είναι οι άδειες που εμφανίζονται πριν την εγκατάσταση μιας εφαρμογής, έτσι ώστε να τις εγκρίνει ο χρήστης. Στην εφαρμογή κρίνονται απαραίτητες οι εξής:

```
<uses-permission android:name="android.permission.INTERNET"></uses-
permission>
```

Είναι η άδεια για το Internet, ώστε ο χρήστης να μπορεί να χρησιμοποιήσει τα δεδομένα του για τη λειτουργία του GPS.

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"></uses-
permission>
```

Η άδεια εύρεσης τοποθεσίας, για να λειτουργήσει το GPS με ακρίβεια.

```
<uses-permission android:name="android.hardware.camera"/>
```

Η άδεια χρήσης της φωτογραφικής μηχανής.

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```



```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Η άδεια εγγραφής και ανάγνωσης της εξωτερικής κάρτας μνήμης, για την αποθήκευση και ανάκτηση της φωτογραφίας που λαμβάνεται.

4.3.4 Το Μενού Ρυθμίσεων και Βοήθειας

Στην Οθόνη Έναρξης έχουν προστεθεί δυο επιλογές στο Μενού. Η μια παραπέμπει στην οθόνη της Βοήθειας, σε περίπτωση που ο χρήστης έχει κάποιες απορίες για τη χρήση της εφαρμογής. Η δεύτερη, παραπέμπει το χρήστη στις ρυθμίσεις τοποθεσίας της συσκευής του για να ενεργοποιήσει το GPS σε περίπτωση που είναι απενεργοποιημένο.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.optionsmenu, menu);
    return true;
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch(item.getItemId()) {
        case R.id.gps_settings:
            startActivityForResult(new
Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS), 0);
            return true;
        case R.id.help:
            startActivity(new Intent("com.ptixiaki.citizencall.HELPSCREEN"));
            return true;
    }
    return false;
}
```

Το layout αυτού του Μενού δεν διαφέρει αρκετά από αυτό της Αρχικής Οθόνης. Γίνεται η εισαγωγή των αντικειμένων και των εικόνων στο αρχείο optionsmenu.xml.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/gps_settings"
        android:title="@string/gps_settings"
        android:icon="@drawable/ic_settings" />

    <item
        android:id="@+id/help"
        android:title="@string/help"
        android:icon="@drawable/ic_help" />
</menu>
```

Κάθε αντικείμενο έχει μια ταυτότητα, έναν τίτλο που ορίζεται με τη χρήση string και μια εικόνα.

4.4 Υλοποίηση της Οθόνης Βοήθειας

Η οθόνη Βοήθειας έχει ως σκοπό την επεξήγηση της λειτουργίας της εφαρμογής στο χρήστη. Αρχικά περιέχει μια αναφορά στο στόχο της εφαρμογής, και έπειτα πληροφορίες για το πώς λειτουργεί. Αποτελείται από ένα layout, το help.xml και μια δραστηριότητα, την helpscreen η οποία περιέχει μόνο τον κώδικα ο οποίος απαιτείται για τον ορισμό του layout. Δεν πραγματοποιούνται άλλες ενέργειες σε αυτή την οθόνη, περιέχονται μόνον οι πληροφορίες που συμπεριλαμβάνονται στο layout.

4.4.1 Το layout help.xml

Η μορφή του layout είναι ScrollView, επειδή οι πληροφορίες καταλαμβάνουν μεγαλύτερο μέγεθος από την οθόνη και απαιτείται κύλιση προς τα κάτω για την εμφάνιση όλων. Μέσα στο ScrollView περιλαμβάνεται το LinearLayout, και μέσα σε αυτό τα αντικείμενα TextView που περιέχουν το κείμενο και ένα ImageView με τη φωτογραφία του Μενού ως υπόδειξη.

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scrollView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/background" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/help"
            android:textAppearance="?android:attr/textAppearanceLarge"/>

        <TextView
            android:id="@+id/text1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/text1" />

        <ImageView
            android:id="@+id/gps_help"
            android:layout_width="fill_parent"
            android:layout_height="0dp"
            android:layout_weight="1"
            android:contentDescription="@string/photo_desc"
            android:src="@drawable/gps_help" />
```

```

<TextView
    android:id="@+id/text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/text2"/>

</LinearLayout>
</ScrollView>

```

4.4.2 Η κλάση `helpscreen.java`

Δεν περιλαμβάνει κάποια επιπλέον λειτουργία πέραν αυτής του ορισμού του layout της δραστηριότητας.

```

public class helpscreen extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.help);
    }
}

```

4.5 Δοκιμές λειτουργίας

Για την εξακρίβωση της ορθής λειτουργίας της εφαρμογής, δεν αρκεί μόνο η δοκιμασία αυτής σε περιβάλλον εξομοίωσης. Πρέπει να λειτουργήσει και σε κανονικές συσκευές ώστε να παρατηρηθούν σφάλματα που μπορεί να προκύψουν. Το Citizen Call έχει δοκιμαστεί και έχει λειτουργήσει άψογα στις παρακάτω συσκευές:

Samsung Galaxy Note 2 - Android 4.4.2

LG G2 - Android 4.4.2

Sony Ericsson Xperia Mini Pro - Android 2.3.4

Samsung Galaxy S Plus - Android 2.3.3

Sony Ericsson X10 Mini - Android 2.3.7

LG G2 - Android 4.4.1

Samsung Galaxy S2 - Android 4.2.3

Asus Nexus 7 - Android 4.4.2

Sony Ericsson Xperia Neo V - Android 2.3.4

4.6 Debugging

Το Debugging (Αποσφαλμάτωση ή Εντοπισμός Σφαλμάτων στην Ελληνική γλώσσα) αποτελεί την σημαντικότερη εκείνη διαδικασία κατά την οποία εξετάζεται ο κώδικας με σκοπό τον εντοπισμό σφαλμάτων που μπορούν να προκαλέσουν προβλήματα στην εκτέλεση και την λειτουργία της εφαρμογής. Είτε πρόκειται για τα μικρότερα δυνατά σφάλματα είτε για ουσιαστικότερα, για παράδειγμα ολοκληρωτική αδυναμία εκτέλεσης του προγράμματος, το πρώτο βήμα της διαδικασίας του Debugging παραμένει το ίδιο.

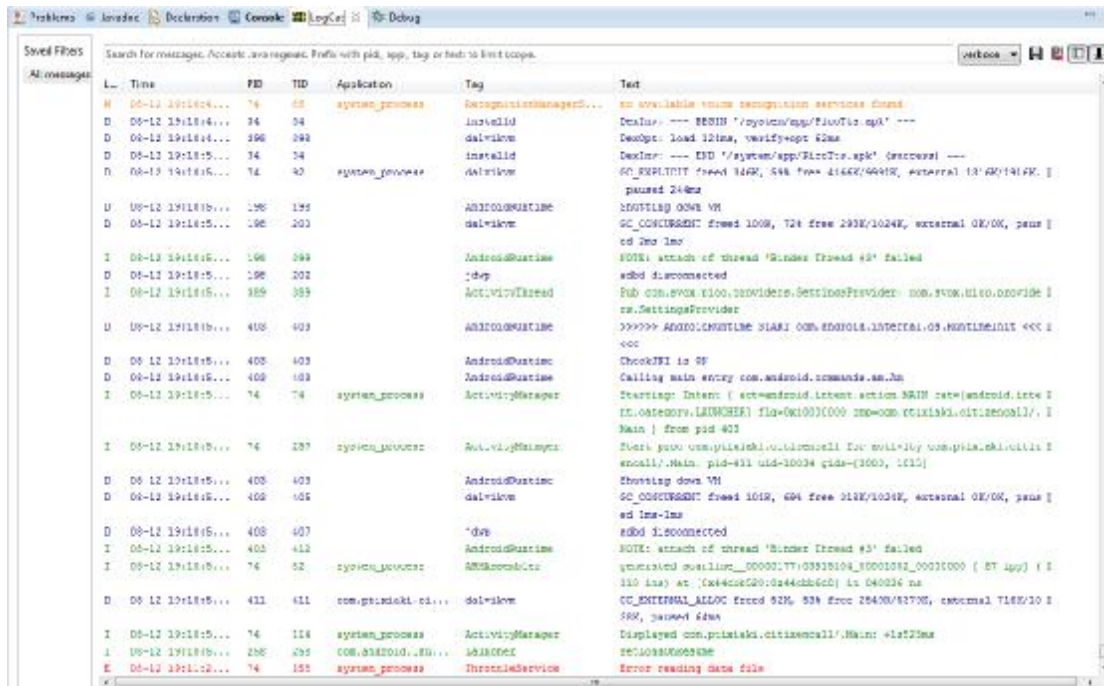
Η αφετηρία ορίζεται πάντοτε ως η επιτυχημένη αναπαραγωγή του προβλήματος. Εν συνεχεία ξεκινά η διαδικασία του εντοπισμού. Το debugging με την χρήση του Eclipse καθίσταται ευκολότερο λόγω της πληθώρας εργαλείων που παρέχονται στον χρήστη. Όπως συμβαίνει με την ανάπτυξη εφαρμογών σε αμιγώς Java περιβάλλοντα, κυρίαρχο ρόλο παίζει το exception handling ή χειρισμός εξαιρέσεων.

Κατά την διαδικασία αυτή ορίζονται εξαιρέσεις για τις τάξεις του προγράμματος από το ίδιο το προγραμματιστικό περιβάλλον ή και επιπρόσθετες από τον προγραμματιστή. Πρόκειται ουσιαστικά για παράλληλες οδούς εκτέλεσης που ακολουθούνται για αντικείμενα που ξεφεύγουν πέρα και έξω από τα εκ προοιμίου οριζόμενα, κοινώς πρόκειται για λάθη. Προγραμματίζοντας σε Android δύσκολα θα παρουσιαστούν σιωπηλά λάθη, η επιδιόρθωση των οποίων είναι εξαιρετικά δύσκολη και χρονοβόρα.

Φυσικά, η καλύτερη μέθοδος και εδώ είναι η πρόληψη. Με δεδομένο πως είναι εξαιρετικά απίθανο να μην παρουσιαστούν σφάλματα στην διάρκεια ανάπτυξης μιας εφαρμογής, αποτελεί προτεραιότητα ο όσο το δυνατόν καθαρός κώδικας με σωστή στοίχιση και συχνές δοκιμές κατά την εισαγωγή νέου κώδικα για επαλήθευση της ορθής λειτουργίας της εφαρμογής.

4.6.1 LogCat

Τρέχοντας την εφαρμογή στον εξομοιωτή ή στο κινητό ενώ είναι συνδεδεμένο με το Eclipse και λειτουργεί το ίδιο σαν εξομοιωτής, παρέχεται η δυνατότητα εμφάνισης του κώδικα στο LogCat, ενός εργαλείου για καταγραφή συμβάντων (εικόνα 4.3).



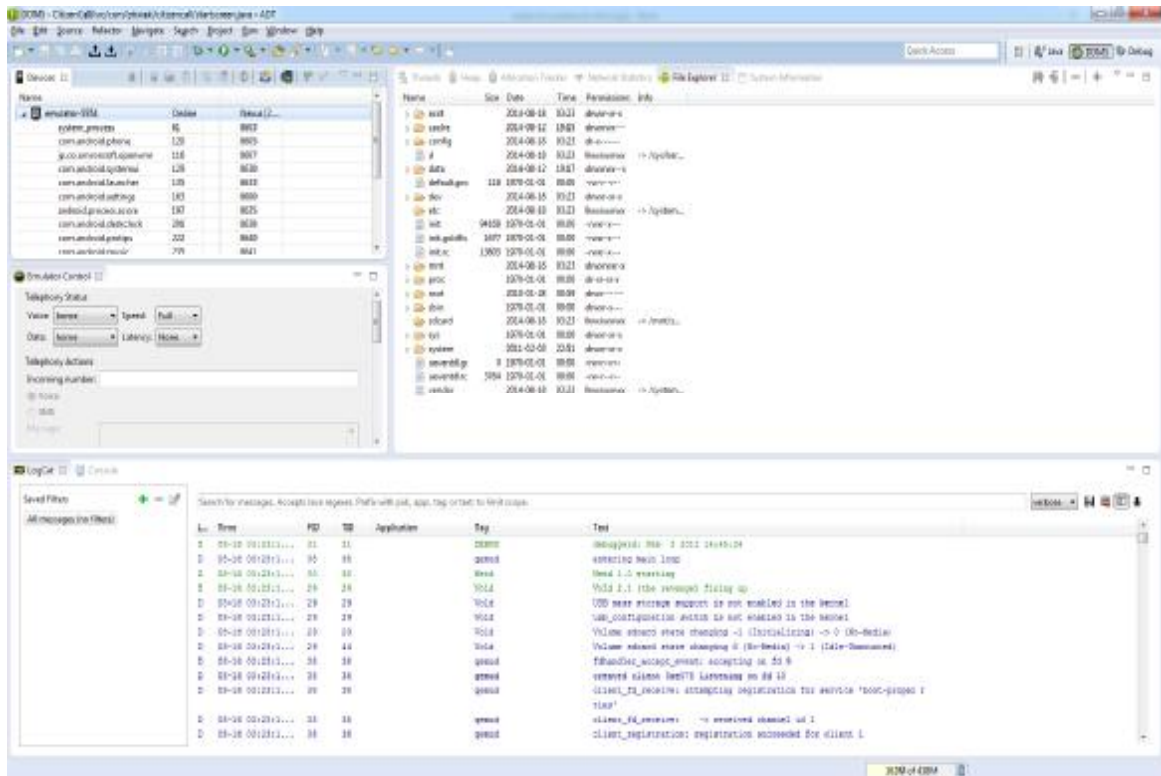
Εικόνα 4.3: Καταγραφή συμβάντων μέσω του LogCat.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, LogCat

Εάν προκύψουν κάποια σφάλματα κατά τη λειτουργία της εφαρμογής, εμφανίζονται με κόκκινα γράμματα στην οθόνη του LogCat με πληροφορίες όπως η κλάση στην οποία εμφανίστηκε το πρόβλημα, και η γραμμή του κώδικα μέσα σε αυτήν.

4.6.2 Dalvik Debug Monitor Server (DDMS)

Ένα ακόμη εργαλείο αποσφαλμάτωσης που συμπεριλαμβάνεται στο Eclipse είναι και ο Dalvik Debug Monitor Server (εικόνα 4.4). Δουλεύει όπως και το LogCat σε εικονικές συσκευές, και παρέχει επιλογές προώθησης θυρών, εισερχόμενες κλήσεις, εικονικά SMS, πληροφορίες τοποθεσίας, το logcat, λήψη εικόνας της οθόνης της συσκευής και άλλα.



Εικόνα 4.4: Η οθόνη πληροφοριών του DDMS.

Πηγή: Στιγμιότυπο κατά τη διάρκεια χρήσης του Eclipse, DDMS

4.7 Τελική έκδοση

Το τελευταίο βήμα πριν την έκδοση μιας εφαρμογής έρχεται μετά από τα στάδια του σχεδιασμού, της υλοποίησης και της αποσφαλμάτωσης. Κατά τη διάρκεια των δοκιμών εντοπίστηκαν και διορθώθηκαν bugs όπως η εξαφάνιση της εικόνας μετά τη λήψη ή το αναγκαστικό κλείσιμο της εφαρμογής κατά την προσπάθεια αποστολής του email δίχως να χει ληφθεί φωτογραφία. Έχοντας μια σταθερή και λειτουργική εφαρμογή, το επόμενο βήμα λοιπόν είναι η δημιουργία του αρχείου αρκ το οποίο μπορεί να εγκατασταθεί σε μια συσκευή. Αυτό προκύπτει με δεξί κλικ επάνω στο πακέτο της εφαρμογής – Android tools – Export Signed Application Package. Κατά τη διάρκεια των βημάτων εξαγωγής του αρχείου, ο προγραμματιστής καλείται να χρησιμοποιήσει ένα κλειδί το οποίο χρησιμοποιείται σαν ηλεκτρονική υπογραφή του πιστοποιητικού της ιδιοκτησίας του. Το Android χρησιμοποιεί το εν λόγω πιστοποιητικό για να προσδιορίσει τον συντάκτη μιας εφαρμογής. Για την εξαγωγή μιας “υπογεγραμμένης” εφαρμογής, απαιτούνται:

- Δημιουργία keystore. Το keystore είναι ένα δυαδικό αρχείο που περιέχει το σύνολο των κλειδιών που δημιουργεί ένας προγραμματιστής στον υπολογιστή του.
- Δημιουργία ιδιωτικού κλειδιού. Ένα κλειδί αντιπροσωπεύει τον συντάκτη που ταυτίζεται με μια εφαρμογή, είτε είναι ένα άτομο είτε μια εταιρία.
- Ανάπτυξη εφαρμογής. Παρέχεται η επιλογή δημιουργίας ενός ανυπόγραφου αρχείου apk της εφαρμογής.
- Υπογεγραμμένη έκδοση της εφαρμογής. Για την χρήση αυτής της επιλογής απαιτείται ένα ιδιωτικό κλειδί. Μετά τη δημιουργία της εφαρμογής, είναι διαθέσιμη προς διανομή ή δημοσίευση στο Google Play.

Συμπεράσματα

Η εκπόνηση της εργασίας τόσο κατά την σύλληψη της αρχικής ιδέας και τον πρώιμο σχεδιασμό της όσο και κατά τα προχωρημένα στάδια πραγμάτωσης και debugging που ακολούθησαν βοήθησε καθοριστικά στην απάντηση του πώς μπορεί ένα σύγχρονο κινητό τηλέφωνο να συνδράμει στην βελτίωση της καθημερινής ζωής του πολίτη. Ο τρόπος με τον οποίο θα γινόταν πραγματικότητα κάτι τέτοιο έπρεπε να βρίσκεται έξω από τα στενά προσχεδιασμένα πλαίσια της μεμονωμένης χρήσης των υποσυστημάτων της συσκευής. Μέσα από την πολύ ενδιαφέρουσα διαδικασία εναρμόνισης της συνδυαστικής λειτουργίας των αισθητήρων λοιπόν, κατορθώνεται και προτείνεται η προώθηση νέων μορφών χρησιμότητας. Η ανάπτυξη μιας εφαρμογής αυτή καθεαυτή είναι μια απαιτητική και χρονοβόρα διαδικασία όμως το αποτέλεσμα ήταν ικανοποιητικό και χρηστικό. Δίχως να επιδιώκεται η πολυπλοκότητα της λειτουργίας κατά την εκτέλεση της εφαρμογής από τον χρήστη, με όσο γίνεται απλά και ξεκάθαρα βήματα, το Citizen Call προτείνεται ως μια αυτοματοποιημένη μέθοδος κοινοποίησής προς τους αρμόδιους φορείς (ή οπουδήποτε αλλού είναι επιθυμητό) των καθημερινών προβλημάτων που συναντώνται στις πόλεις.

Μελλοντικά υπάρχουν τρόποι και ιδέες εξέλιξης της εφαρμογής. Αυτοί μπορεί να είναι η ενσωμάτωση των χαρτών του Google Maps για την ένδειξη της τρέχουσας τοποθεσίας του χρήστη, ή η αποστολή αιτήματος με τη χρήση server ώστε να μην χρειάζεται η χρήση email.

Όπως έχει δείξει η ιστορία της ανάπτυξης των κινητών συσκευών, με την πάροδο του χρόνου εμφανίζονται όλο και περισσότερες ευκολίες προς το χρήστη αλλά και προς τους προγραμματιστές, για να παράγουν περισσότερο στοχευμένες, ευκολόχρηστες και επωφελείς εφαρμογές.

Βιβλιογραφία

- Klemens G., 2010, *The Cellphone: The History and Technology of the Gadget that Changed the World*, McFarland & Company Inc. Publishers, Jefferson, North Carolina, and London.
- Agar J., 2004, *Constant Touch: A Global History of the Mobile Phone*, Totem Books.
- Misra A. and Dubey A., 2013, *Android Security: Attacks and Defenses*, Auerbach Publications; 1 edition.
- Gargenta M., 2011, *Learning Android*, O'Reilly Media; 1 edition.
- Murphy M., 2010, *Beginning Android*, Apress; 1 edition.
- Cinar O., 2012, *Android Apps with Eclipse*, Apress; 1 edition.
- Friesen J., 2010, *Learn Java for Android Development*, Apress; 1 edition.
- Spaanenburg L. and Spaanenburg H., 2010, *Cloud Connectivity and Embedded Sensory Systems*, Springer.

- Η επίσημη ιστοσελίδα του Android, <http://developer.android.com/>
- Η επίσημη ιστοσελίδα του Eclipse, <https://www.eclipse.org/>
- Η ιστοσελίδα του stackoverflow, μιας κοινότητας που παρέχει βοήθεια σε θέματα προγραμματισμού, <http://stackoverflow.com/>
- Industry Leaders Announce Open Platform for Mobile Devices, http://www.openhandsetalliance.com/press_110507.html με πρόσβαση στις 15/2/2014.
- Google's Android parts ways with Java industry group, <http://www.cnet.com/news/googles-android-parts-ways-with-java-industry-group/> με πρόσβαση στις 28/2/2014.
- T-Mobile G1: Full Details of the HTC Dream Android Phone, <http://gizmodo.com/5053264/t+mobile-g1-full-details-of-the-htc-dream-android-phone> με πρόσβαση στις 28/2/2014.
- API - application program interface, <http://www.webopedia.com/TERM/A/API.html> με πρόσβαση στις 1/3/2014

- History of Android OS. From beta till Lime Pie, <http://forum.gsmhosting.com/vbb/f831/history-android-os-beta-till-lime-pie-1683166/>, με πρόσβαση στις 1/3/2014.
- Android Evolution – Android Beta to Jelly Bean, <http://www.spectrose.com/android-history-all-versions.html>, με πρόσβαση στις 1/3/2014.
- Dashboards, <https://developer.android.com/about/dashboards/index.html>, με πρόσβαση στις 17/3/2014.
- Microsoft tablet PC, <http://msdn.microsoft.com/en-us/library/ms840465.aspx>, με πρόσβαση στις 19/5/2014.
- Difference between Tablet and Smartphone, <http://www.differencebetween.info/difference-between-tablet-and-smartphone>, με πρόσβαση στις 19/5/2014.
- Watch The Incredible 70-Year Evolution Of The Cell Phone, <http://www.businessinsider.com/complete-visual-history-of-cell-phones-2011-5/1938-1#1999-18>, με πρόσβαση στις 8/6/2014.
- The Evolution of Cell Phone Design Between 1983-2009, <http://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/>, με πρόσβαση στις 8/6/2014.
- Cellular Network, <http://www.telecomabc.com/c/cellular.html>, με πρόσβαση στις 10/6/2014.
- How Mobile Networks Work, <http://www.quepublishing.com/articles/article.aspx?p=2021961>, με πρόσβαση στις 10/6/2014.
- GSM – Architecture, http://www.tutorialspoint.com/gsm/gsm_architecture.htm, με πρόσβαση στις 16/6/2014.
- GSM Network Architecture, http://www.radio-electronics.com/info/cellulartelecomms/gsm_technical/gsm_architecture.php, με πρόσβαση στις 16/6/2014.
- History of GSM and More, <http://www.ldpost.com/telecom-articles/History-of-GSM-and-More.html>, με πρόσβαση στις 16/6/2014.

- Αισθητήρας επιτάχυνσης,
<http://physicsandroid.wordpress.com/%CE%B1%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82-%CE%B5%CF%80%CE%B9%CF%84%CE%AC%CF%87%CF%85%CE%BD%CF%83%CE%B7%CF%82/>, με πρόσβαση στις 31/7/2014.
- Learn About The Sensors Used In Smartphones (Very Easy Language),
<http://techntoday.blogspot.gr/2013/04/learn-about-sensors-used-in-smartphones.html>, με πρόσβαση στις 31/7/2014.
- iPhone 4 Gyroscope Teardown,
<https://www.ifixit.com/Teardown/iPhone+4+Gyroscope+Teardown/3156>, με πρόσβαση στις 31/7/2014.
- Αισθητήρας εγγύτητας,
<http://physicsandroid.wordpress.com/%CE%B1%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82-%CE%B5%CE%B3%CE%B3%CF%8D%CF%84%CE%B7%CF%84%CE%B1%CF%82/>, με πρόσβαση στις 31/7/2014.