



# 2014

## Ανάπτυξη Εφαρμογής Ταξίμετρου σε Android



Πανεπιστήμιο Πειραιά

Τμήμα Ψηφιακών Συστημάτων

Μεταπτυχιακό Πρόγραμμα Σπουδών:

«Ψηφιακές Επικοινωνίες και Δίκτυα»

Ανάπτυξη Εφαρμογής Ταξίμετρου σε  
πλατφόρμα Android.

Περδίκης Ανδρέας

ΜΕ 10069

## Περίληψη

Στην παρούσα εργασία παρουσιάζεται το λογισμικό Android για φορητές συσκευές και η ανάπτυξη μιας εφαρμογής ταξίμετρου σε αυτή την πλατφόρμα. Στο πρώτο κεφάλαιο γίνεται μια περιγραφή του λειτουργικού συστήματος Android καλύπτοντας θέματα αρχιτεκτονικής και σχεδίασης, αναλύοντας τα διακριτά συστήματα που το αποτελούν. Θα αναφερθούν θέματα όπως ο πυρήνας του Linux που στηρίζεται το Android, οι Βιβλιοθήκες (API's) για διεπαφή με λειτουργίες όπως τα γραφικά 2D και 3D, η βάση δεδομένων SQLite και άλλα. Επίσης θα δούμε την εικονική μηχανή του Dalvik όπου τρέχει κάθε διεργασία, το χρόνο εκτέλεσης Εφαρμογής και το πλαίσιο εφαρμογής Android. Επιπλέον στο ίδιο κεφάλαιο θα αναλυθούν τα δομικά μέρη μιας εφαρμογής, τα βασικά αρχεία της, και τα περιεχόμενα των φακέλων του. Τέλος περιγράφεται πώς λειτουργεί η ασφάλεια των εφαρμογών στην πλατφόρμα android.

Στο δεύτερο κεφάλαιο παρουσιάζεται ο κυριότερος ανταγωνισμός στην σχετική αγορά και καταδεικνύονται οι δυνατότητες που πλέον μπορεί να έχουν οι εφαρμογές για smart phones.

Στο τρίτο κεφάλαιο ερευνούμε τα εργαλεία ανάπτυξης λογισμικού Android, αναλύοντας εκτενέστερα το Android Software Developer Kit (**SDK**), τον εξομοιωτή android το Android Native Development Kit (NDK) καθώς και άλλα εργαλεία ανάπτυξης εφαρμογών. Επιπλέον, στο κεφάλαιο αυτό παρουσιάζεται μια εφαρμογή τιμολόγησης της διαδρομής για χρήση σε ταξί(Ταξίμετρο) εντός Αττικής. Η εφαρμογή περιέχει όλες τις πιθανές τιμολογήσεις που προβλέπονται ανάλογα τον χρόνο και τον τόπο αναχώρησης(Λιμάνια, Αεροδρόμια κ). Για να αναπτυχθεί η εφαρμογή χρειάστηκε να ξεπεραστούν δύο μεγάλα εμπόδια. Το πρώτο εμπόδιο είναι η συμβατότητα με τις περισσότερες αν όχι όλες τις συσκευές. Για το λόγο αυτό χρησιμοποιήθηκαν οι βιβλιοθήκες υποστήριξης (**Android Support Libraries**) του Android. Οι βιβλιοθήκες αυτές παρέχουν εκδόσεις των Android API's με backward compatibility. Δεύτερο μεγάλο εμπόδιο ήταν η υποστήριξη χάρτη σε εφαρμογή android. Για να ξεπεραστεί και το δεύτερο εμπόδιο χρησιμοποιήθηκε η τεχνολογία των **Google Play Services** και το API της Google για **Google Maps**. Με τα **Google Play Services** μπορούμε να χρησιμοποιούμε τις καινούριες υπηρεσίες της Google όπως το **Activity recognition** για την εύρεση της θέσης μας, και με το **Google Maps** για την εμφάνιση της στο χάρτη. Έτσι δημιουργούμε εφαρμογή όπου ο χρήστης μπορεί να βλέπει ανά πάσα στιγμή την θέση του στον χάρτη ακόμα και σε χώρους όπου το GPS δεν είναι διαθέσιμο.

## Περιεχόμενα

Περιεχόμενα .....	0
Πίνακας Εικόνων .....	3
Εισαγωγή.....	6
Κεφάλαιο 1: Περιγραφή του Λειτουργικού Συστήματος Android.....	8
1.1 Περιγραφή του Λειτουργικού Συστήματος Android.....	8
1.2. Αρχιτεκτονική Android.....	17
1.2.1 Πυρήνας Linux (Linux Kernel).....	17
1.2.2 Βιβλιοθήκες.....	17
1.2.3 Η εικονική μηχανή Dalvik .....	18
1.2.4 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime) .....	19
1.2.5 Πλαίσιο Εφαρμογής (Application Framework).....	19
1.3 Στο Εσωτερικό μιας Εφαρμογής Android.....	20
1.3.1 Το αρχείο AndroidManifest.xml .....	21
1.3.2 Οι φάκελοι src & res .....	21
1.3.3 Οι υπόλοιποι φάκελοι του project.....	21
1.3.4 Δομικά Μέρη μιας Εφαρμογής.....	22
1.4 Ασφάλεια στο Android .....	22
Κεφάλαιο 2: Ανταγωνισμός .....	24
2.1 Windows Mobile .....	24
2.2 SYMBIAN .....	28
2.3 IOS .....	31
Κεφάλαιο 3: Ανάπτυξη της Εφαρμογής .....	33
3.1 Εργαλεία ανάπτυξης λογισμικού.....	33
3.1.1 Android SDK .....	33

3.1.2 Ο εξομοιωτής (emulator).....	34
3.1.3 Άλλα εργαλεία του Android.....	35
3.1.4 Android NDK (Native Development Kit) .....	36
3.2 Σχεδίαση της Εφαρμογής.....	40
3.2.1 Πιθανοί Χρήστες .....	40
3.2.2 Απαιτήσεις.....	40
3.2.3 Λειτουργίες .....	40
3.2.4 User Interface .....	43
3.3 Υλοποίηση της Εφαρμογής .....	46
3.3.1 Περιβάλλον Ανάπτυξης Eclipse .....	46
3.3.2 Google Maps API .....	46
3.4 Κυριότερα Αρχεία .....	52
3.4.1 AndroidManifest.xml .....	52
3.4.2 Activity.xml .....	53
3.4.3 Strings.xml .....	53
3.4.4 Activity.java .....	53
Κεφάλαιο 4: Χρήση της Εφαρμογής.....	54
Επίλογος.....	57
Παράρτημα: Κώδικας Κυριοτέρων αρχείων.....	58
Αναφορές .....	78

### **Πίνακας Εικόνων**

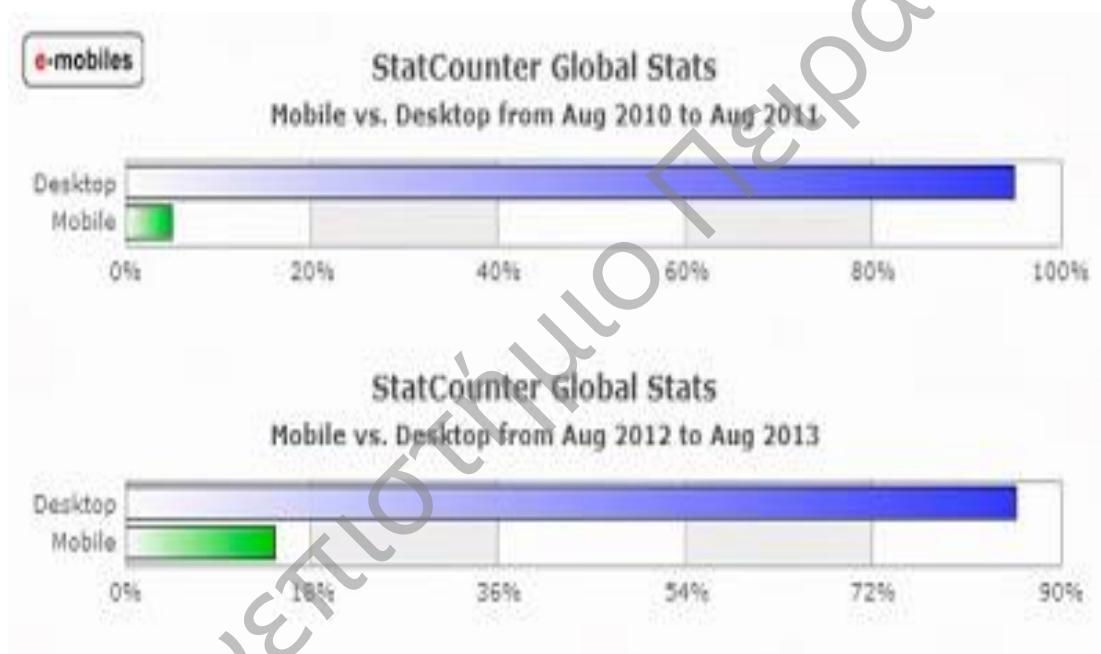
Εικόνα 1: Χρήση smart phones για σύνδεση σε κοινωνικά δίκτυα .....	6
Εικόνα 2: Χαρακτηριστικό menu σε συσκευή Android .....	9
Εικόνα 3: Αρχιτεκτονική Android .....	12
Εικόνα 4: Λογότυπα Εκδόσεων Android .....	13
Εικόνα 5: Χρήση Εκδόσεων του Λειτουργικού Android ανά Έκδοση.....	16

Εικόνα 6: Activity Manager Lifecycle .....	20
Εικόνα 7: Χαρακτηριστική οθόνη Windows Phone .....	24
Εικόνα 8: Χαρακτηριστική οθόνη symbian.....	28
Εικόνα 9: Αρχιτεκτονική Symbian .....	29
Εικόνα 10: Χαρακτηριστική οθόνη ios .....	31
Εικόνα 11: Αρχιτεκτονική ios.....	32
Εικόνα 12: Διάγραμμα ροής για την καταγραφή της διανυθείσας διαδρομής.....	41
Εικόνα 13: Διάγραμμα ροής για τον υπολογισμό του κόστους διαδρομής	42
Εικόνα 14: Διάγραμμα Καταστάσεων Εφαρμογής .....	43
Εικόνα 15: User Interface για την καταχώρηση των στοιχείων μίσθωσης .	44
Εικόνα 16: User Interface για την λειτουργία της καταγραφής της διαδρομής και του κόστους της .....	45
Εικόνα 17: Χρήση Google API (1) .....	47
Εικόνα 18: Χρήση Google API (2) .....	48
Εικόνα 19: Χρήση Google API (3) .....	49
Εικόνα 20: Χρήση Google API (4) .....	49
Εικόνα 21: Χρήση Google API (5) .....	50
Εικόνα 22: Χρήση Google API (6) .....	50
Εικόνα 23: Χρήση Google API (7) .....	51
Εικόνα 24: Βασική Δομή Εφαρμογής Android ( <a href="http://osarena.net/android/aguides/mathenontas-to-android-androidmanifest-xml-is-all-about-mathimata-4-5-6.html">http://osarena.net/android/aguides/mathenontas-to-android-androidmanifest-xml-is-all-about-mathimata-4-5-6.html</a> ).....	52
Εικόνα 25: Εικόνα Επιλογής Μίσθωσης (1) .....	54
Εικόνα 26: Εικόνα Επιλογής Μίσθωσης (1)	<b>Error! Bookmark not defined.</b>

Πανεπιστήμιο Πειραιώς

## Εισαγωγή

Η εξέλιξη των κινητών τηλεφώνων και των smart phones τα τελευταία χρόνια ήταν αλματώδης. Το γεγονός αυτό σε συνδυασμό με την επίσης ραγδαία μείωση του απαιτούμενου κόστους απόκτησης των συσκευών αυτών, επέφερε έκρηξη στην χρήση τους. Σήμερα είναι πολύ μεγάλο το ποσοστό των ανθρώπων όλων των ηλικιών που χρησιμοποιεί τις προηγμένες υπηρεσίες των παρόχων κινητής τηλεφωνίας εκμεταλλευόμενοι τις απεριόριστές δυνατότητες των συσκευών. Χαρακτηριστικό είναι το παρακάτω γράφημα που φανερώνει το πόσο αυξάνεται το ποσοστό που χρησιμοποιεί smart phones προκειμένου να απόκτηση πρόσβαση σε υπηρεσίες κοινωνικής δικτύωσης.



Εικόνα 1: Χρήση smart phones για σύνδεση σε κοινωνικά δίκτυα

Η εξέλιξη των συσκευών κινητής τηλεφωνίας από το 1973 έως σήμερα πέρασε από το στάδιο όπου οι συσκευές δεν είχαν κάποιο λειτουργικό σύστημα και περιορίζονταν με απλό μενού, λιγότες δυνατότητες, μονόχρωμες οθόνες και πληκτρολόγιο με μεγάλα πλήκτρα. Χαρακτηριστικό τους ήταν επίσης μεγάλος όγκος και οι εξωτερικές κεραίες που τα καθιστούσε άβολα για μεταφορά. Στην συνέχεια οι συσκευές έγιναν μικρότερες και ελαφρύτερες. Οι κεραίες έγιναν εσωτερικές και απέκτησα προηγμένες δυνατότητες όπως απεικόνιση πληροφοριών σε έγχρωμες οθόνες, λήψη και παρουσίαση φωτογραφιών και

video, αναπαραγωγή ραδιοφώνου, σύνδεση στο Internet κα [1]. Συνοπτικά η εξέλιξη αυτή έχει ως εξής:

- Παρά το γεγονός ότι τα κινητά έχουν καταλάβει τη σημερινή κοινωνία, στη πραγματικότητα η τεχνολογία προϋπήρχε από το 1940. Αρχικά η τεχνολογία κυψελών χρησιμοποιήθηκε κυρίως στα αστυνομικά αυτοκίνητα και σε ορισμένα οχήματα έκτακτης ανάγκης και είναι η ίδια που χρησιμοποιείτε και σήμερα.
- Η πρώτη γενιά (1G) κινητών που ήταν διαθέσιμη για το κοινό ήταν το 1983 από τη Motorola. Ήταν αναλογικά και λιγότερο αξιόπιστα από τα σημερινά ψηφιακά. Είχαν δυνατότητα μόνο για φωνητικές κλήσεις.
- Το 1990 φτιάχτηκαν τα κινητά δεύτερης γενιάς (2G), που επικοινωνούσαν ψηφιακά, πιο γρήγορα και με λιγότερο θόρυβο. Μειώθηκε ο όγκος των συσκευών και των μπαταριών τους καθώς επίσης βελτιώθηκε και η ενεργειακή τους απόδοση.
- Σήμερα τα κινητά τρίτης γενιάς (3G), που έχουν πιο πολλές δυνατότητες, όπως για παράδειγμα, EMAIL, SMS, πλοήγηση στο Διαδίκτυο, κλπ. Όλα αυτά τα καθιστούν σαν ένα υπολογιστή τσέπης και επομένως φτάνουμε στο σημείο να μιλάμε για Λειτουργικό Σύστημα Κινητών Τηλεφώνων.

Με την εμφάνιση των λειτουργιών αυτών, παρουσιάστηκε η ανάγκη αυτές να συντονίζονται με λειτουργικό σύστημα στα πρότυπα των ηλεκτρονικών υπολογιστών. Με τον όρο Λειτουργικό Σύστημα Κινητού Τηλεφώνου, νοείται ένα σύστημα που ελέγχει μια συσκευή και τις πληροφορίες αυτής και είναι παρόμοιο με το Λειτουργικό Σύστημα ενός Υπολογιστή. Τα κυριότερα λειτουργικά συστήματα για κινητά τηλέφωνα που κυκλοφορούν στην αγορά είναι:

- Google Android
- Apple ios
- Windows Mobile
- Symbian OS

Η υπόλοιπη εργασία είναι δομημένη ως εξής:

Στο κεφάλαιο 1 γίνεται μία σύντομη επισκόπηση του λειτουργικού συστήματος Android. Στο κεφάλαιο 2 γίνεται μία αναφορά στα κυριότερα λειτουργικά συστήματα του ανταγωνισμού. Στο κεφάλαιο 3 παρουσιάζεται η διαδικασία ανάπτυξης της εφαρμογής. Η διαδικασία ανάπτυξης περιλαμβάνει την φάση της σχεδίασης και την φάση της υλοποίησης. Στην τελευταία παρουσιάζονται τα εργαλεία που χρησιμοποιούν συχνότερα οι προγραμματιστές προκειμένου να αναπτύξουν τέτοιες εφαρμογές. Στο Κεφάλαιο 4 παρουσιάζεται το πώς μπορεί να χρησιμοποιηθεί η εφαρμογή που αναπτύχτηκε και το τι δυνατότητες έχει [2].



## Κεφάλαιο 1: Περιγραφή του Λειτουργικού Συστήματος Android

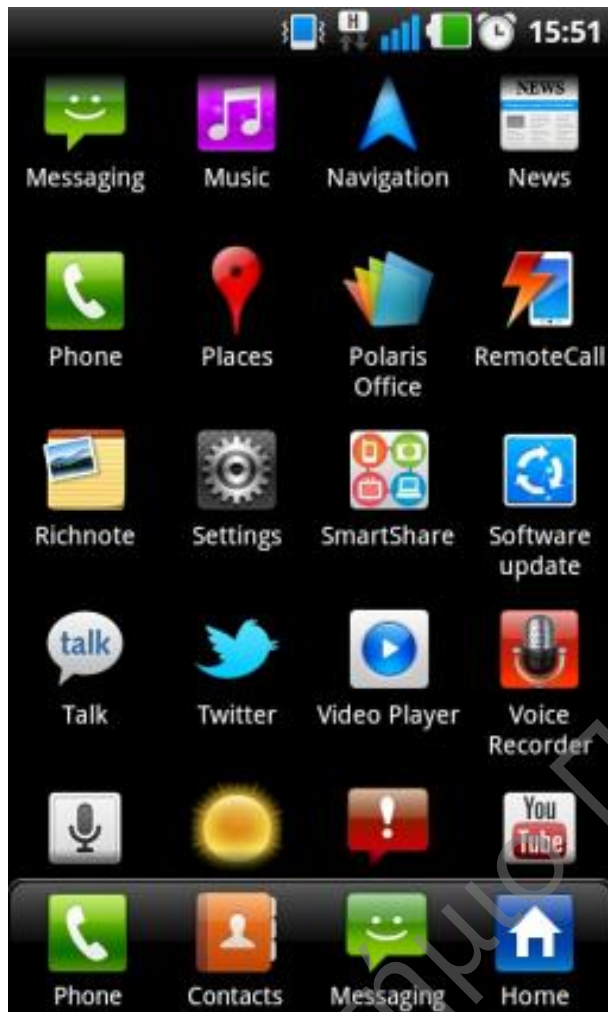
### 1.1 Περιγραφή του Λειτουργικού Συστήματος Android

Το Android είναι ένα σύνολο προγραμμάτων για smart phones, που περιλαμβάνει:

- ένα λειτουργικό σύστημα,
- middleware
- εφαρμογές.

Αναπτύχθηκε από την Google για έξυπνες συσκευές κινητής τηλεφωνίας. Η ονομασία του είναι Ελληνική και σημαίνει ανδροειδές – ανθρωποειδές. Η ανάπτυξη του ήταν και παραμένει σε υψηλά επίπεδα. Σήμερα χρησιμοποιείται από το μεγαλύτερο ποσοστό των κατασκευαστών συσκευών. Προκαλεί το ενδιαφέρον μεγάλου επίσης ποσοστού προγραμματιστών για smart phones με αποτέλεσμα να αναπτύσσονται συνεχώς μεγάλος αριθμός ανάλογων εφαρμογών. Σημαντικό ρόλο σε αυτό διαδραματίζει η απόλυτη συμβατότητα του με τις υπηρεσίες της Google καθώς και το ότι οι μεγάλο πλήθος εφαρμογών του και υπηρεσιών του παρέχονται εντελώς δωρεάν.

Το Software Development Kit του παρέχει τα εργαλεία και τα απαραίτητα APIs, για την ανάπτυξη εφαρμογών που θα μπορούν να τρέξουν υπό το λειτουργικό σύστημα Android, σε γλώσσα Java. Με αυτό η Google, κατάφερε να καταλάβει την πρώτη θέση στις προτιμήσεις των χρηστών smart phones. Πρόκειται για πακέτο λογισμικού που σχεδιασμένο να δημιουργεί τον τρόπο επικοινωνίας με τις δυνατότητες της συσκευής[3].



Εικόνα 2: Χαρακτηριστικό μενού σε συσκευή Android

Τα βασικά χαρακτηριστικά της εμπειρίας του χρήστη μίας συσκευής που τρέχει το λειτουργικό σύστημα Android είναι:

- Λειτουργίες Οθόνης
- Αποθήκευση Δεδομένων (Με χρήση Βάσεων Δεδομένων SQLite)
- Συνδεσιμότητα (WiFi – Bluetooth)
- Μηνύματα
- Internet (Με χρήση φυλλομετρητή τεχνολογίας WebKit)
- Πολυμέσα (Εικόνες, Ήχος, Βίντεο)
- Υποστήριξη Υλικού (Κάμερες, Οθόνη Αφής, GPS, Αισθητήρες Επιτάχυνσης, Μαγνητόμετρα, 2D και 3D Επιταχυντές Γραφικών)
- Εφαρμογές (Εγκατάσταση με χρήση WIFI από το PlayStore)

- Υποστήριξη Java (Η εικονική μηχανή Dalvik που έχει το Android είναι μια εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένη για χρήση σε φορητές συσκευές. Παρόλο που δεν είναι μια πρότυπη εικονική μηχανή Java, μπορεί να μεταγλωττίσει και να εκτελέσει εφαρμογές Java)
- Οθόνη Αφής Πολλαπλών Σημείων (Το Android από την αρχή είχε τη δυνατότητα Multi-Touch, όμως είχε κλειδωθεί μέχρι πρόσφατα σε επίπεδο πυρήνα εξαιτίας της πατέντας της Apple στις τεχνολογίες οθονών αφής. Στις καινούριες εκδόσεις Android είναι βασικό χαρακτηριστικό)

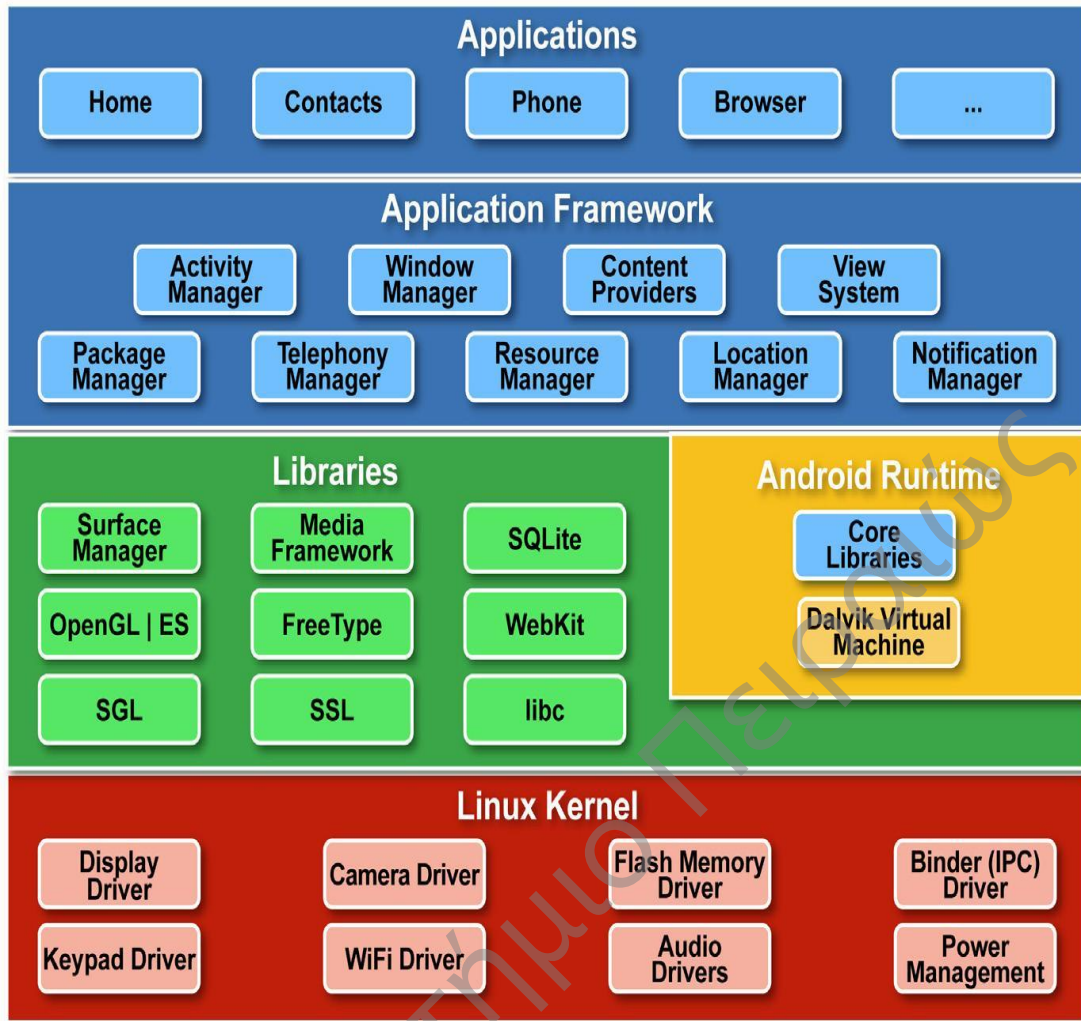
Η αρχιτεκτονική του συντίθεται από τέσσερα επίπεδα. Τα επίπεδα αυτά είναι τα εξής:

- Πυρήνας Linux (Linux Kernel): Το Android βασίζεται σε πυρήνα Linux για τις χαμηλού επιπέδου υπηρεσίες του. Οι υπηρεσίες αυτές περιλαμβάνουν:
  - διαχείριση των προγραμμάτων οδήγησης της συσκευής,
  - αποδοτική χρησιμοποίηση της μνήμης,
  - χρονοπρογραμματισμό των διεργασιών,
  - εγκατάσταση και διατήρηση της δικτύωσης με την χρήση όλων των διαθέσιμων σχετικών τεχνολογιών (GEM, HSDPA, WiFi, Bluetooth).
- Βιβλιοθήκες (Libraries): Στο επίπεδο αυτό περιλαμβάνονται οι εγγενείς βιβλιοθήκες και οι εκείνες χρόνου εκτέλεσης.
  - Εγγενείς Βιβλιοθήκες (Internal): Οι βιβλιοθήκες που χρησιμοποιούν οι λειτουργίες του έχουν αναπτυχθεί σε γλώσσα προγραμματισμού C/C++ και χρησιμοποιούνται με κατάλληλες διεπαφές της Java.
  - Βιβλιοθήκες Χρόνου Εκτέλεσης (Runtime): Στο επίπεδο αυτό κατατάσσονται οι βασικές βιβλιοθήκες της Java που χρησιμοποιούνται και η εικονική μηχανή Dalvik η οποία είναι μία υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές που αναπτύχθηκε από την Google. Χαρακτηριστικό της είναι το γεγονός της χρήσης των πιο συνεκτικών τύπων αρχείων που παρουσιάζονται πιο αποδοτικά σε περιβάλλον συσκευών με περιορισμένους φυσικούς πόρους.
- Πλαίσιο Υποστήριξης Εφαρμογών (Application Framework): Στο επίπεδο αυτό βρίσκονται οι λειτουργικές μονάδες που χρησιμοποιούνται για την ανάπτυξη εφαρμογών. Υπάρχει ένα σύνολο από βασικές προεγκατεστημένες λειτουργίες που στην συνέχεια μπορεί να επεκταθούν από τον κατασκευαστή της

έξυπνης συσκευής. Τα κυριότερα στοιχεία του επιπέδου αυτού είναι:

- View System: Είναι το σύστημα εκείνο που παρέχει τα στοιχεία των γραφικών διεπαφών (πχ buttons, textboxes, links).
- Content Providers: Είναι οι λειτουργικές μονάδες εκείνες που επιτρέπουν στις εφαρμογές την πρόσβαση στα δεδομένα άλλων εφαρμογών ή την διάθεση των δικών τους δεδομένων σε άλλες.
- Resource Manager: Ως πόροι αναφέρονται όλα εκείνα τα στοιχεία ενός προγράμματος που δεν είναι κώδικας. Η μονάδες αυτές ρυθμίζουν την πρόσβαση σε αυτά.
- Notification Manager: Διαχειρίζεται τα μηνύματα που ανταλλάσσονται μεταξύ των εφαρμογών ή εκείνων που οι εφαρμογές παράγουν.
- Activity Manager: Ελέγχει τις δραστηριότητες της εφαρμογής κατά την λειτουργία της και διατηρεί μιας στοίβα που επιτρέπει την πλοήγηση στην πρότερες καταστάσεις των διεπαφών της.
- Location Manager: Δομική μονάδα που διατηρεί την θέση της συσκευής ανά πάσα στιγμή.
- Επίπεδο Εφαρμογών (Application): Στο ανώτερο επίπεδο βρίσκονται οι εφαρμογές που χειρίζεται ο τελικός χρήστης. Βασικό τους χαρακτηριστικό είναι η διαφανής χρήση των λειτουργιών και των υπηρεσιών των κατώτερων επιπέδων της αρχιτεκτονικής αποκρύπτοντας τις λεπτομέρειες υλοποίησης τους από τους τελικούς χρήστες [4].

Στο παρακάτω σχήμα φαίνεται σχηματικά η αρχιτεκτονική του Android.



Εικόνα 3: Αρχιτεκτονική Android

Οι εκδόσεις του έχουν χαρακτηριστικό το ότι περιγράφονται από το όνομα ενός γλυκού και ακολούθησαν την επόμενη εξέλιξη [6].



Εικόνα 4: Λογότυπα Εκδόσεων Android

- **CupCake 1.5**

Η έκδοση “Cupcake”, βασισμένη στο Linux Kernel 2.6.27, παρουσιάστηκε στις 30 Απριλίου του 2009. Υποστηρίζει νέες λειτουργίες για την κάμερα τις συσκευής, όπως η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα απευθείας από το τηλέφωνο. Έχει νέο έξυπνο πληκτρολόγιο με πρόβλεψη κειμένου. Υποστηρίζει πρότυπο Bluetooth A2DP και AVRCP ενώ έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.

- **Donut 1.6**

Η έκδοση “Cupcake”, βασισμένη στο Linux Kernel 2.6.27, παρουσιάστηκε στις 30 Απριλίου του 2009. Υποστηρίζει νέες λειτουργίες για την κάμερα τις συσκευής, όπως η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα απευθείας από το τηλέφωνο. Έχει νέο έξυπνο πληκτρολόγιο με πρόβλεψη κειμένου. Υποστηρίζει πρότυπο Bluetooth A2DP και AVRCP ενώ έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.

- **Eclair 2.0 – 2.1**

Η έκδοση “Eclair”, βασισμένη και αυτή στον Linux Kernel 2.6.29, παρουσιάστηκε στις 26 Οκτωβρίου του 2009, ενώ τον Ιανουάριο του 2010 επανεκδόθηκε σε Android 2.1 Eclair (MR1). Σε αυτή την έκδοση υπάρχει ακόμα ταχύτερη απόκριση

του υλικού σε σχέση με τις δυο προηγούμενες και πλέον υποστηρίζονται περισσότερες οθόνες και αναλύσεις. Υπάρχει νέος browser ο οποίος υποστηρίζει το πρότυπο HTML5, νέο User Interface, και βελτιωμένοι χάρτες Google (Google Maps 3.1.2). Έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom. Επίσης έχει βελτιωθεί η κλάση MotionEvent ώστε να υπάρχει η δυνατότητα για γεγονότα πολλαπλής αφής (multitouch events). Υποστηρίζεται Bluetooth 2.1 και έχει βελτιωθεί και το πληκτρολόγιο.

- **Froyo 2.2**

Η έκδοση “Froyo”, βασισμένη στο Linux Kernel 2.6.32, παρουσιάστηκε στις 20 Μαΐου του 2010. Υπάρχουν βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση. Έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον browser, υπάρχει πλέον Adobe Flash 10.1, ενώ υποστηρίζεται καλύτερα πλέον το Microsoft Exchange. Έχει γίνει ανανέωση του Android Market. Ο χρήστης μπορεί πλέον να ελέγχει αν θα γίνεται ή όχι κίνηση πακέτων δεδομένων από το δίκτυο κινητής τηλεφωνίας. Υπάρχει η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Επίσης το τηλέφωνο πλέον μπορεί να μετατραπεί σε WiFi hotspot.

- **GingerBread 2.3 – 2.3.7**

Η έκδοση “Gingerbread”, βασισμένη στο Linux Kernel 2.6.35.7, παρουσιάστηκε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Στην έκδοση αυτή υπάρχουν αλλαγές στο User Interface το οποίο έχει γίνει πιο απλό και ταχύ, ενώ υποστηρίζονται πλέον οθόνες μεγάλων μεγεθών και αναλύσεων. Υπάρχει πλέον το πρωτόκολλο SIP για κλήσεις μέσω VoIP, υποστηρίζεται ο τύπος βίντεο WebM/VP8 και ο κωδικοποιητής AAC, έχει βελτιωθεί ο ήχος καθώς και οι λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών. Υπάρχει η δυνατότητα για Copy-Paste σε όλο το σύστημα και όχι μόνο στην ίδια εφαρμογή. Υποστηρίζεται το NFC (Near Field Communication) και η ύπαρξη πολλαπλών καμερών. Επίσης, έχει βελτιωθεί η ενεργειακή υποστήριξη και έχει γίνει μετάβαση από το σύστημα αρχείων YAFFS στο ext4 στις νέες συσκευές.

- **HoneyComb 3.0 – 3.2**

Η έκδοση “Honeycomb”, βασισμένη στο Linux Kernel 2.6.36, παρουσιάστηκε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets. Οι αλλαγές που έγιναν στην έκδοση αυτή έχουν να κάνουν κυρίως με τη βελτίωση της υποστήριξης των tablets. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρρηνοι και τετραπύρρηνοι επεξεργαστές. Ακόμα, έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου (recent apps) να περνάει από μια εφαρμογή σε άλλη. Υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk καθώς η ανάγνωση βιβλίων μέσω του Google eBooks. Επιπλέον, μπορούν να κρυπτογραφηθούν όλα τα δεδομένα χρήστη.

- **IceCreamSandwich 4.0**

Η έκδοση “Ice Cream Sandwich”, βασισμένη στο Linux Kernel 3.0.1, παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Για άλλη μια φορά έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Πλέον στο User Interface, το οποίο είναι και πάλι διαφορετικό, υπάρχουν εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν στις συσκευές. Βελτίωση της ασφάλεια του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων. Η εφαρμογή Android Beam αξιοποιεί πλέον το NFC αφού επιτρέπει την αποστολή δεδομένων από τη συσκευή σε όσες βρίσκονται εντός μιας μικρής ακτίνας εμβέλειας. Ακόμα με την ύπαρξη του Wi-Fi Direct συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς την μεσολάβηση κάποιου access point. Τέλος, υποστηρίζεται η εγγραφή βίντεο σε 1080p.

- JellyBean 4.1 – 4.3

Η έκδοση “Jelly Bean”:

- 4.1 με Linux Kernel 3.0.31, Ιούνιος 2012
- 4.2 με Linux Kernel 3.4.0, Οκτώβριος 2012
- 4.3 με Linux Kernel 3.4.39, Ιούλιος 2013

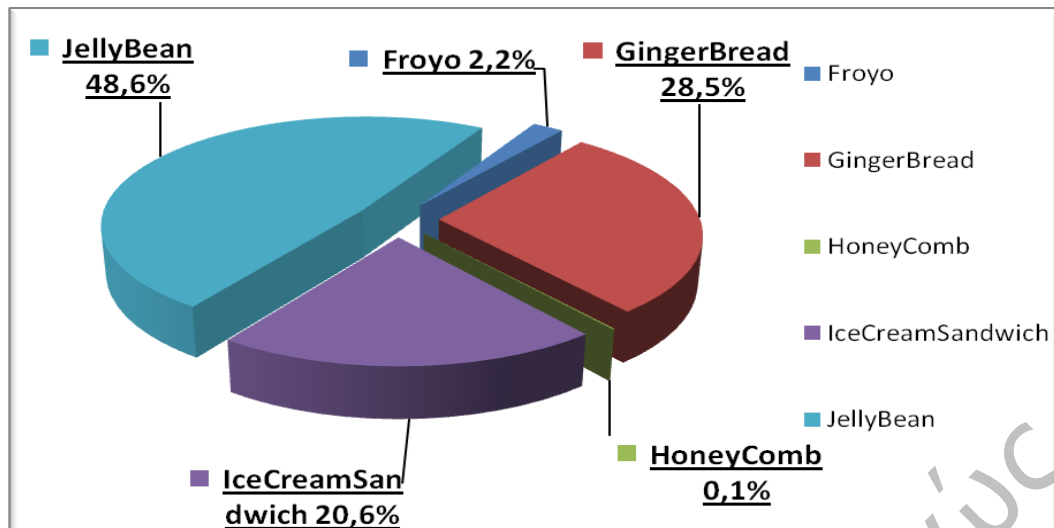
Στις εκδόσεις αυτές προστέθηκαν χαρακτηριστικά όπως το **Dial pad autocomplete**, όπου ο χρήστης μπορεί να τηλεφωνήσει αναζητήσει τηλέφωνα πατώντας τους αριθμούς του πληκτρολογίου και το android προτείνει επαφές-τηλέφωνα. Το **Bluetooth AVRCP** όπου πλέον ο χρήστης μπορεί να συγχρονίσει το Bluetooth του με ηχοσυστήματα αυτοκινήτου ή σπιτιού, το **Android Beam**, όπου ο χρήστης μπορεί να διαμοιράζεται τα δεδομένα του, φωτογραφίες και video μέσω NFC, τα **Restricted profiles**, το **Notification bar**, το **OpenGL ES 3.0** για καλύτερα γραφικά καθώς και νέο User Interface για την εφαρμογή της κάμερας. [14]

- **Kit Kat 4.4**

Η έκδοση “Kit Kat 4.4” με Linux Kernel 3.4, ξεκίνησε τον Σεπτέμβρη 2013. Στην καινούρια αυτή έκδοση ήρθαν καινούριες δυνατότητες όπως καλύτερη διαχείριση μνήμης, με αποτέλεσμα να τρέχει και σε συσκευές με 512 MB RAM. Επίσης αρκετά χαρακτηριστικά βελτιώθηκαν όπως το Google Now και η full screen. Επιπλέον υπάρχει ενσωματωμένη η δυνατότητα απευθείας αποθήκευσης στο **Cloud** χωρίς να χρειάζεται να αποθηκεύσει ο χρήστης στο κινητό του, καθώς και η δυνατότητα εκτύπωσης μέσω WiFi ή Bluetooth. Τέλος, προστέθηκε η υποστήριξη για δύο νέους αισθητήρες, μετρητής βημάτων και αισθητήρας βήματος, που βρήκε χρήση στις εφαρμογές μέτρησης τρεξίματος. [15]

Ο βαθμός χρήσης της κάθε έκδοσης αποτυπώνεται στο ακόλουθο γράφημα.





Εικόνα 5: Χρήση Εκδόσεων του Λειτουργικού Android ανά Έκδοση

Τα πλεονεκτήματα του Android πηγάζουν από την αρχιτεκτονική του. Στο κατώτερο επίπεδο της αρχιτεκτονική του βρίσκεται ο πυρήνας Linux ο οποίος δεν είναι ιδιαίτερα απαιτητικός σε φυσικούς πόρους. Το χαρακτηριστικό αυτό του επιτρέπει να εγκαθίσταται σε διαφορετικές πλατφόρμες υποβόσκοντος υλικού αλλά και να παρουσιάζει υψηλό βαθμό επεκτασιμότητας ώστε μελλοντικά να είναι ικανό να προσαρμοστεί σε νέες απαιτήσεις. Η χρήση πυρήνα Linux επίσης είναι ένας παράγοντας που καθιστά το λειτουργικό σύστημα πιθανών αξιόπιστο. Σε βάθος χρόνου έχουν αναπτυχθεί και λειτουργήσει αξιόπιστα απαιτητικές εφαρμογές βασιζόμενες σε Linux χρήση από κινητές συσκευές αμβλύνοντας έτσι την αβεβαιότητα λόγω της βραχείας διάρκειας ζωής του μέχρι σήμερα. Η ανάπτυξη των εφαρμογών του γίνεται σε γλώσσα προγραμματισμού Java της οποίας το βασικότερο χαρακτηριστικό είναι ότι και αυτή δεν εξαρτάται από την πλατφόρμα του υλικού όπου τρέχουν τα προγράμματα της. Η φύση των εφαρμογών που αναπτύσσονται για Android έχει παρουσιάζει σε υψηλό βαθμό προσαρμοστικότητα σε διαφορετικά είδη συσκευών. Επιπλέον παρέχει όλα τα εργαλεία για την αποδοτική διασύνδεση των συσκευών που υποστηρίζει είτε με ασύρματα είτε με ασύρματα φυσικά μέσα. Παρέχει δωρεάν πρόσβαση στον κώδικα του ώστε να μπορούν τρίτοι προγραμματιστές να ενσωματώσουν πρόσθετες λειτουργίες, να επαναπροσδιορίσουν τον τρόπο εκτέλεσης δεδομένων λειτουργιών του. Προσαρμόζεται σε διεθνή πρότυπα λειτουργίας για κάθε επίπεδο της αρχιτεκτονική του και παρέχει την δυνατότητα για δωρεάν χρήση, ανάπτυξη και δημοσίευση εφαρμογών. Τέλος όσο αφορά την πρόσθετη απαίτηση των έξυπνων συσκευών κινητής τηλεφωνίας (σε σχέση με εκείνες της τρίτης γενιάς), την ασφάλεια, παρέχει ικανοποιητικό επίπεδο ασφάλειας αφού ο πυρήνας Linux μπορεί να ελέγχει τις δικές του διαδικασίες αλλά και αυτές των εφαρμογών ανάλογα με τα δικαιώματα πρόσβασης που έχουν στους πόρους του συστήματος. Επιπροσθέτως παρέχει και διαδικασίες ελέγχου για τον περιορισμό των επιτρεπόμενων ενεργειών για κάθε μία διαδικασία που εκτελείται.

Συνοψίζοντας, τα βασικά πλεονεκτήματα του Android πηγάζουν από τα χαρακτηριστικά του:

- Ανοιχτή Πλατφόρμα: Με βάση το Linux, παρέχει πρόσβαση σε βασικές λειτουργίες της κινητή συσκευή
- Ξεπερνά τεχνικά όρια: Το Android σας επιτρέπει να συνδυάζετε τις πληροφορίες από το Web, λειτουργίες του κινητού σας σε συνδυασμό με τα δεδομένα των εταιρικών εφαρμογών και πληροφοριών σας.
- Απλοποιημένη Ανάπτυξη: τόσο οι πελάτες όσο και προγραμματιστές ζητούν μειωμένους κύκλους ανάπτυξης και εύκολα στη χρήση σχετικά εργαλεία. Η Linux-based πλατφόρμα Android επιτρέπει την πλήρη πρόσβαση σε όλη την λειτουργικότητα των κινητών συσκευών, δεδομένου ότι βασίζεται σε ένα ισχυρό κινητήρα Java.

## 1.2. Αρχιτεκτονική Android

### 1.2.1 Πυρήνας Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης
- Οδηγό Wifi και Bluetooth
- Οδηγό κάμερας
- κλπ.

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 4-5 χρόνια ακόμα.

### 1.2.2 Βιβλιοθήκες

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά

ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. Μερικές από τις κύριες βιβλιοθήκες του Android είναι:

- **System C library** – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.

- **Βιβλιοθήκες Πολυμέσων** – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG

- **Surface Manager** – διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.

- **LibWebCore** – μια μοντέρνα μηχανή υποστήριξης πλοήγηση στο διαδίκτυο (browser engine) η οποία χρησιμοποιείτε και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.

- **SGI** – η γνωστή μηχανή δισδιάστατων γραφικών

- **Βιβλιοθήκες 3D** – μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.

- **FreeType** – παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.

- **SQLite** – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων.

### 1.2.3 Η εικονική μηχανή Dalvik

Σχεδόν το σύνολο των APIs του Android βασίζονται στη γλώσσα προγραμματισμού Java. Στην Java ως γνωστόν υπάρχει η λεγόμενη Java Virtual Machine στην οποία εκτελείτε ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει κάτι παρόμοιο και δεν είναι άλλο από την εικονική μηχανή Dalvik.

Η Dalvik λοιπόν είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android. Η κάθε εφαρμογή τρέχει μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα. Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία **.dex** ώστε να τρέξουν στην VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση.

Το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο. Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

### 1.2.4 Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime)

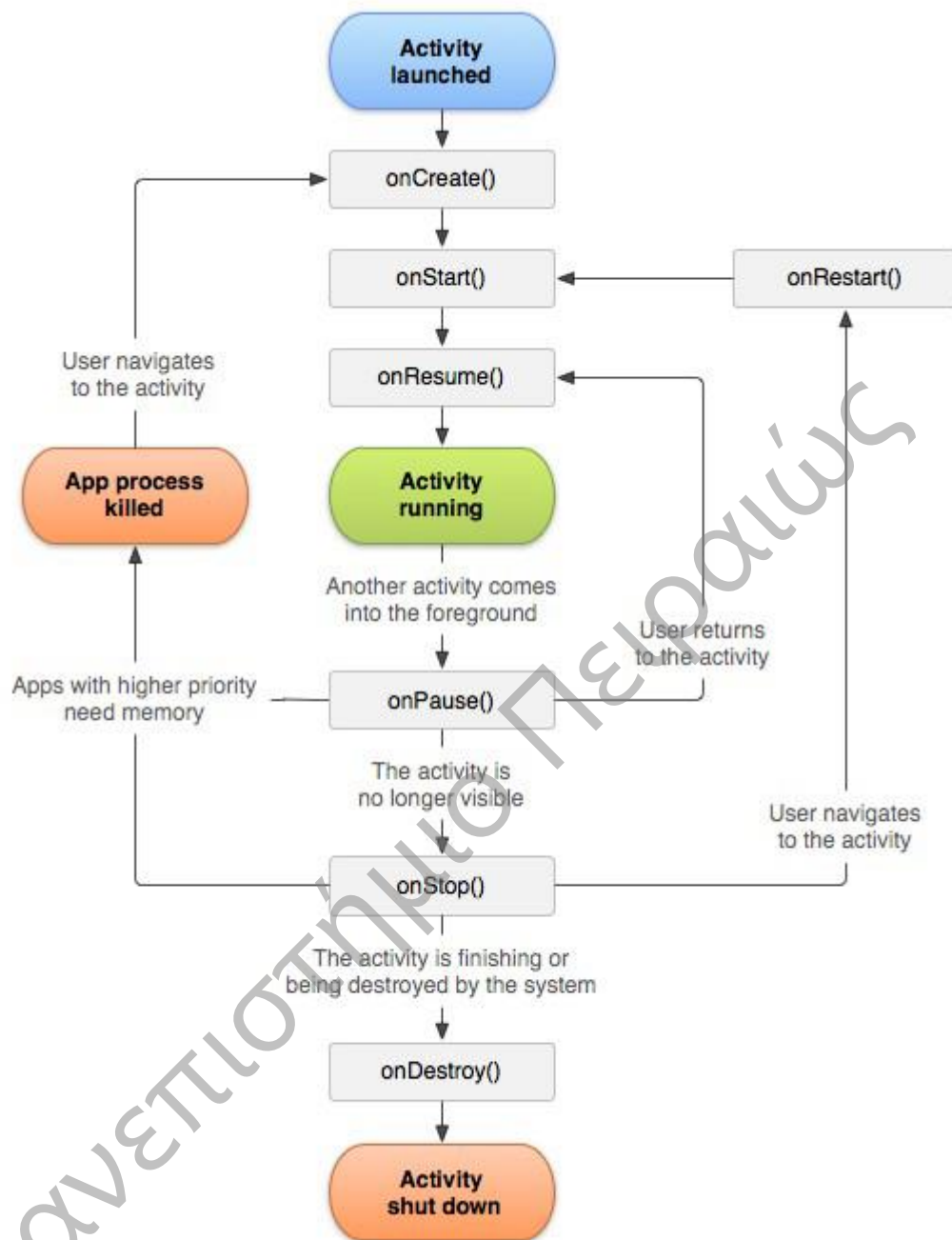
Ο χρόνος εκτέλεσης των εφαρμογών του Android, βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Εδώ βρίσκουμε το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της εικονικής μηχανής Dalvik τις λειτουργίες τις οποίες, περιγράψαμε παραπάνω.

### 1.2.5 Πλαίσιο Εφαρμογής (Application Framework)

Το Android παρέχει στους developers μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης και τη δυνατότητα να αναπτύξουν με αυτή ιδιαίτερα καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι developers έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκήνιου, και πάρα πολλές ακόμη δυνατότητες οι οποίες βασίζονται στα APIs που είναι διαθέσιμα.

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **Σύστημα προβολών (View System)** – αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ.
- **Πάροχος Περιεχομένου (Content Provider)** – δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- **Διαχειριστής Πόρων (Resource Manager)** – παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.
- **Διαχειριστής Ειδοποιήσεων (Notification Manager)** – δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ.
- **Διαχειριστής Δραστηριοτήτων (Activity Manager)** – διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών. Στην παρακάτω εικόνα φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας.



Εικόνα 6: Activity Manager Lifecycle

### 1.3 Στο Εσωτερικό μιας Εφαρμογής Android

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να εγκαταστήσουμε στις συσκευές μας.

Ξεκινώντας, η κάθε εφαρμογή αποτελείται όπως είπαμε από πολλά αρχεία δομημένα σε φακέλους. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name) το οποίο χρησιμοποιείτε από το λειτουργικό σύστημα για

αναγνώριση της εφαρμογής. Μια εφαρμογή μπορεί να αποτελείται από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας της εφαρμογής, αλλά μόνο από ένα κύριο.

### 1.3.1 Το αρχείο AndroidManifest.xml

Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομα του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη
- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ., που περιέχει και χρησιμοποιεί η εφαρμογή.

Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο συστατικό κάθε εφαρμογής.

### 1.3.2 Οι φάκελοι src & res

Στον φάκελο src (εκ του source) περιέχονται τα αρχεία κλάσης τις Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία, κλπ. Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας. Ο φάκελος res (εκ του resources) περιέχει όλα τα αρχεία εικόνας, κειμένου, xml layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο src. Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος drawable ο οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή μας, ο φάκελος layout ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος values στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται στην εφαρμογή.

### 1.3.3 Οι υπόλοιποι φάκελοι του project

Ένα project αποτελείται από περισσότερους από τους 3 βασικούς φακέλους, κάποιος από τους οποίους μπορεί να θεωρηθούν και περιττοί αναλόγως την περίπτωση. Στο project λοιπόν περιλαμβάνονται και ο φάκελος με τα διαθέσιμα APIs αναλόγως την έκδοση που έχουμε επιλέξει να δουλέψουμε, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο build path του project μας, και επίσης περιλαμβάνει και τις διαβαθμίσεις του φακέλου res, όπως είναι οι φάκελοι drawable-hdpi, drawable-mdpi, layout-port, menu, κλπ. Σε αυτούς περιλαμβάνονται τα ειδικά διαμορφωμένα αρχεία πόρων που έχουμε τοποθετήσει ώστε να είναι διαθέσιμα από το λειτουργικό σύστημα, αναλόγως την περίπτωση.

### 1.3.4 Δομικά Μέρη μιας Εφαρμογής

Παραπάνω αναφέραμε ότι όλα τα δομικά μέρη της εφαρμογής πρέπει να αναφέρονται αναλυτικά στο αρχείο `AndroidManifest.xml`, ποια είναι όμως αυτά τα δομικά μέρη και πια η λειτουργία του καθενός:

- **Δραστηριότητες (Activities)** – Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.
- **Προθέσεις (Intents)** – Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.
- **Υπηρεσίες (Services)** – Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματά ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.
- **Πάροχος Περιεχόμενου (Content Providers)** - Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent, ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.
- **Δέκτες Μετάδοσης (Broadcast Receivers)** – Πρόκειται για ένα είδους υπηρεσία η οποία αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνει να ενημερώσει το σύστημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθώς μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

### 1.4 Ασφάλεια στο Android

Τη στιγμή που μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στο δικό της εικονική μηχανή η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης
- Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό ID το οποίο είναι άγνωστο στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής, και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.

- Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινείται μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείται πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του Android Manifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να τις παρέχει πρόσβαση στους πόρους που ζητάει.

Πανεπιστήμιο Πειραιώς



## Κεφάλαιο 2: Ανταγωνισμός

### 2.1 Windows Mobile

Το WINDOWS MOBILE είναι ένα λογισμικό για Smartphone και rocketPC τηλεφώνων που αναπτύχθηκε από την Microsoft. Βασίζεστε στον πυρήνα 5.2 των Windows C.E. και εμφανίστηκε για πρώτη φορά το 2000. Έχει σχεδιαστεί με τέτοιο τρόπο ώστε να έχει παρόμοια χαρακτηριστικά και εμφάνιση με τις εκδόσεις των Windows για υπολογιστές. Οι προγραμματιστές μπορούν να αναπτύξουν εφαρμογές για το Windows Mobile, χωρίς περιορισμούς που επιβάλλονται από την Microsoft. Μερικές εφαρμογές είναι διαθέσιμες για αγορά από το Windows Marketplace [9].



Εικόνα 7: Χαρακτηριστική οθόνη Windows Phone

Στις περισσότερες συσκευές υπάρχει μία γραφίδα, η οποία χρησιμοποιείται για την εισαγωγή εντολών, με πάτημα στην οθόνη.

Βασικά χαρακτηριστικά τους είναι τα παρακάτω:

- Η αρχική οθόνη (today screen) η οποία δείχνει τις πληροφορίες όπως τις πληροφορίες κατόχου, την τρέχουσα ημερομηνία, τα μηνύματα ηλεκτρονικού ταχυδρομείου κ.α.

- Η γραμμή εργασιών η οποία δείχνει την τρέχουσα ώρα και την ένταση του ήχου.
- Το Office Mobile, Windows Media Player και Internet Explorer τα οποία είναι αντίστοιχα με αυτά των Windows για PC.
- Δυνατότητα διαμοίρασης μια σύνδεσης Internet σε Η/Υ μέσω Bluetooth και θηρών USB.
- Έχει σύστημα αρχείων παρόμοιο με των Windows και υποστηρίζει αρχεία πολλών τύπων.

Το μερίδιο των συσκευών με WINDOWS MOBILE στην αγορά smartphone μειώνεται από έτος σε έτος, όπως στον πίνακα. Αυτό οφείλεται στην εισαγωγή στην αγορά των λειτουργικών συστημάτων Google Android και i-OS.

Η αρχική έκδοση του Windows Mobile παρουσιάστηκε το 2000 με την ονομασία Rarier. Τα χαρακτηριστικά της έκδοσης αυτής είχαν ως εξής:

- Ανάλυση 240x320
- Pocket Office (Word, Excel, Outlook)
- Pocket Internet Explorer
- Pocket Media Player
- Microsoft Reader
- Microsoft Money
- Υποστήριξη γραπτών και ηχητικών σημειώσεων
- Υποστήριξη αναγνώρισης χαρακτήρων
- Αποστολή και λήψη αρχείων μέσω υπερύθρων

Το 2001 παρουσιάστηκε το λειτουργικό σύστημα Pocket PC με την ονομασία Merlin το οποίο χρησιμοποιήθηκε και σε Smartphones. Τα επιπλέον χαρακτηριστικά του περιλάμβαναν:

- Προστέθηκε στο Pocket Word ορθογραφικός έλεγχος και εργαλείο μέτρησης των λέξεων
- Προστέθηκε το πρωτόκολλο WAP (Wi-Fi Protected Access)
- και η δυνατότητα κατεβάσματος στο Pocket Internet Explorer
- Υποστήριξη Virtual Private Network (VPN)
- Δυνατότητα συγχρονισμού φακέλων
- MSN Messenger
- Terminal Services
- Windows Media Player 8 με δυνατότητα Streaming
- Microsoft Reader 2

Το 2003 Διατέθηκε η έκδοση Windows Mobile 2003 που πρόσθεσε τα εξής χαρακτηριστικά:

- Βελτιωμένη διεπαφή με τον χρήστη με τη διαχείριση συσκευής μέσω Bluetooth

- Αποστολή κ λήψη αρχείων μέσω Bluetooth
- Υποστήριξη ακουστικών Bluetooth
- Εφαρμογή προβολής , περικοπής και αποστολής μέσω email, των εικόνων
- Προσθήκη του παιχνιδιού Jawbreaker
- Windows Media Player 9 με βελτιωμένο Streaming
- Επιλογή απάντησης σε SMS
- Δυνατότητα χρήσης αρχείων ήχου, ως ήχος κλήσης
- Το 2004 κυκλοφόρησε η έκδοση Windows Mobile 2003 CE. Τα χαρακτηριστικά της έκδοσης αυτής ήταν:
- Δυνατότητα αλλαγής φαντού αρχικής οθόνης
- Ενιαία διάταξη στήλης Pocket Internet Explorer
- Υποστήριξη των αναλύσεων οθόνης 640x480, 176x220, 240x240, 480x480
- Υποστήριξη του πρωτοκόλλου Wi-Fi Protected Access (WPA)

Το 2005 παρουσιάστηκε η έκδοση Windows Mobile 5.0. Αρχικά η Microsoft θα παρείχε υποστήριξη σε αυτή τη συσκευή μέχρι τον Οκτώβριο του 2010, όμως επέκτεινε την υποστήριξη μέχρι το 2015. Οι συσκευές Windows Mobile 5.0 παρουσιάζουν αύξηση στην ζωή μπαταρίας. Χρησιμοποιούν την μνήμη RAM σε συνδυασμό με την μνήμη Flash, ως πρωταρχικό μέσο αποθήκευσης, σε αντίθεση με τις παλαιότερες συσκευές. Τα προγράμματα κ η πρόσβαση στα δεδομένα τρέχουν στη RAM και αποθήκευση τους, καθώς και τα αντίγραφα ασφαλείας γίνονται στην Flash.

Το 2007 παρουσιάζεται το Windows Mobile 6.0, στο οποίο υπήρξαν πολλές αναθεωρήσεις μέχρι το 2010. Χαρακτηριστικά αυτού ήταν:

- Υποστηρίζει τις αναλύσεις οθόνης 320x320 και 800x480
- Υποστηρίζει το Office Mobile από τα smartphones
- Βελτιωμένη απομακρυσμένη σύνδεση
- Δυνατότητα κρυπτογράφησης κάρτας μνήμης
- Βελτιωμένη κοινή χρήση του Internet
- Υποστήριξη JavaScript στο Internet

Το Windows Phone διαδέχθηκε το Windows Mobile το 2010. προσανατολισμένο για προσωπική χρήση περισσότερο παρά για επιχειρησιακή. Τα βασικά του χαρακτηριστικά είναι:

- Διαθέτει ένα νέο περιβάλλον χρήστη. Η αρχική οθόνη, ονομάζεται πλέον οθόνη Έναρξης και αποτελείται από κινούμενα εικονίδια. Τα εικονίδια είναι συνδέσεις με εφαρμογές και οι χρήστες μπορούν να τα αφαιρέσουν να τα αναδιατάξουν ή να προσθέσουν νέα. Τα εικονίδια αυτά έχουν δυνατότητα άμεσης ανανέωσης (π.χ. το εικονίδιο ενός λογαριασμού email θα εμφανίσει τον αριθμό των μη αναγνωσμένων μηνυμάτων).
- Χρησιμοποιεί τεχνολογία πολλαπλής αφής.
- Διαθέτει εικονικό πληκτρολόγιο το οποίο έχει την δυνατότητα εισαγωγής emoticons, διαθέτει ορθογραφικό έλεγχο και πρόβλεψη λέξεων.
- Ο Internet Explorer του Windows Phone επιτρέπει στον χρήστη να διατηρεί μια λίστα αγαπημένων ιστοσελίδων στην οθόνη έναρξης που οδηγούν σε αυτές. Υποστηρίζει έως και 6 καρτέλες τις οποίες μπορεί να φορτώσει παράλληλα. Επίσης μπορεί να αποθηκεύσει εικόνες από ιστοσελίδες, να διαμοιράσει ιστοσελίδες μέσω email και να ψάξει λέξεις ή φράσεις σε μια ιστοσελίδα.
- Διαχειρίζεται τα μηνύματα μέσω νημάτων. Τα νήματα επιτρέπουν στο χρήστη να συνδυάσει τις επαφές του με το Windows Live Messenger, το Facebook chat καθώς και τα μηνύματα κειμένου. Μπορεί επίσης να μετατρέψει την ομιλία σε γραπτό μήνυμα και αντίστροφα.
- Οι επαφές μπορούν να προστεθούν από τον χρήστη ή να εισαχθούν από το Facebook και το Twitter.
- Υποστηρίζει πάρα πολλές μορφές εικόνας, ήχου και βίντεο.
- Μπορεί να συνδεθεί με το X-Box Live και ο χρήστης να αλληλεπιδράσει με το avatar του στο X-Box.
- Υποστηρίζει πολλά προφίλ Bluetooth.
- Διαθέτει το κουμπί αναζήτησης. Πατώντας το, ενώ μια εφαρμογή είναι ανοιχτή, αναζητά πληροφορίες μέσα στην εφαρμογή. Αν δεν είναι καμία εφαρμογή ανοιχτή, ανοίγει την εφαρμογή Bing η οποία είναι η προεπιλεγμένη μηχανή αναζήτησης[9].

## 2.2 SYMBIAN

Το Symbian OS είναι ένα λειτουργικό σύστημα για κινητά τηλέφωνα, το οποίο δημιουργήθηκε από την Symbian Ltd με την γλώσσα προγραμματισμού C++ και σήμερα συντηρείται από την Accenture. Υποστηρίζει πολλαπλές διεργασίες και «τρέχει» αποκλειστικά σε επεξεργαστές AMR. Χρησιμοποιείται από πολλούς κατασκευαστές συσκευές κινητών τηλεφώνων (κυρίως από την Nokia). Στο τέλος του δεύτερου τριμήνου του 2010 τα κινητά με λειτουργικό Symbian ήταν 385 εκατομμύρια. Οι πρώτες εκδόσεις του Symbian OS βασίζονταν στον πυρήνα EKA1, ο οποίος σήμερα έχει αντικατασταθεί από τον ισχυρότερο και σταθερότερο EKA2. Ξεχωριστό χαρακτηριστικό του είναι το γεγονός ότι έχει δημιουργηθεί ειδικά για φορητές συσκευές, κάτι που πρακτικά σημαίνει ότι έχει δημιουργηθεί με γνώμονα τη χαμηλή κατανάλωση ενέργειας, αλλά και πόρων συστήματος όπως επεξεργαστική ισχύς και μνήμη RAM. Επιτρέπει επίσης την εγκατάσταση πρόσθετων εφαρμογών, οι οποίες θεωρούνται εφάμιλλες σε δυνατότητες με αυτές των ηλεκτρονικών υπολογιστών.



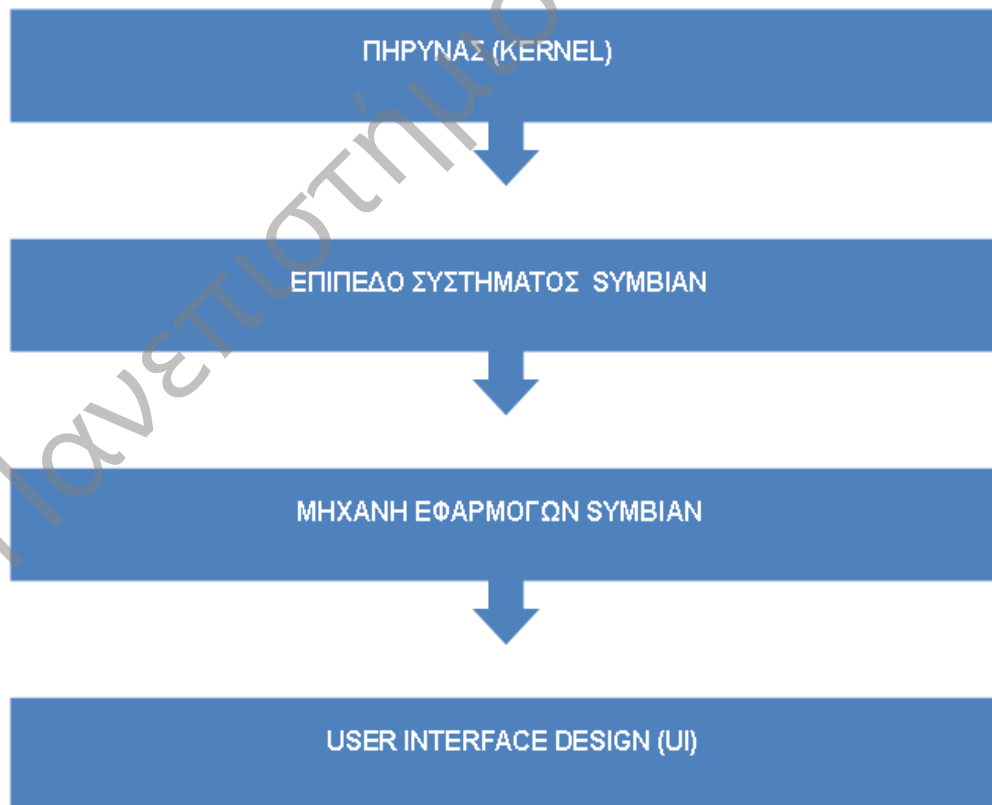
Εικόνα 8: Χαρακτηριστική οθόνη symbian

Το λειτουργικό σύστημα Symbian, μπορεί να «τρέξει» σε συσκευές με διαφορετικό μέγεθος οθόνης. Το μέγεθος της κάθε συσκευής και οι εισόδους που παρέχει στον χρήστη (πληκτρολόγιο, οθόνη αφής κ.α.) η κάθε συσκευή, ασκούν επίδραση στις εφαρμογές γιατί εξαρτώνται τόσο από το μέγεθος της οθόνης όσο και από τις εισόδους αυτές.

Τα επίπεδα που περιλαμβάνει η αρχιτεκτονική του λειτουργικού συστήματος Symbian από κάτω προς τα πάνω είναι τα παρακάτω:

- Ο πυρήνας (kernel): Η λειτουργία του πυρήνα είναι να διαχειρίζεται τους πόρους του υλικού, όπως η μνήμη RAM και οι συσκευές υλικού και να εξασφαλίζει και να ελέγχει την πρόσβαση των εφαρμογών στους πόρους αυτούς. Η κεντρική μονάδα επεξεργασίας λειτουργεί πάντα υπό τις οδηγίες του πυρήνα. Το επίπεδο αυτό δεν είναι άμεσα προσβάσιμο από τις εφαρμογές του χρήστη.
- Επίπεδο συστήματος Symbian: Αυτό το επίπεδο περιλαμβάνει τα περισσότερα APIs (Application Programming Interface) του λειτουργικού, τα οποία προσφέρουν λειτουργίες όπως ο χειρισμός αλφαριθμητικών και ο χρονοπρογραμματισμός των εφαρμογών.
- Μηχανή Εφαρμογών Symbian: Στο επίπεδο αυτό υπάρχει μια μηχανή η οποία επιτρέπει στον χρήστη να έχει πρόσβαση στα δεδομένα των ενσωματωμένων λειτουργιών ή εφαρμογών. Τέτοιες εφαρμογές ή λειτουργίες είναι ο τηλεφωνικός κατάλογος, το ημερολόγιο κ.α.
- User Interface Design (UI): Το UI είναι το τέταρτο και τελευταίο επίπεδο, και είναι ο σχεδιασμός της επικοινωνίας μεταξύ του χρήστη και τις συσκευής.

Σχηματικά η αρχιτεκτονική του παρουσιάζεται στο ακόλουθο διάγραμμα



Εικόνα 9: Αρχιτεκτονική Symbian

Τα βασικά του χαρακτηριστικά συνοψίζονται στα εξής:

- Διαδικασίες και νήματα: Είναι ένα πολλαπλών καθηκόντων και πολύπλοκο λειτουργικό σύστημα. Πολλές διαδικασίες μπορούν να τρέξουν ταυτόχρονα, μπορούν να επικοινωνήσουν η μια με την άλλη, και μπορούν να αξιοποιήσουν τα πολλαπλά νήματα που τρέχουν εσωτερικά σε κάθε διαδικασία.
- Κοινό σύστημα υποστήριξης αρχείων: Οργανώνει την πρόσβαση στον αποθηκευτικό χώρο του συστήματος, χρησιμοποιώντας ένα πρότυπο σύστημα αρχείων, όπως και τα μεγάλα λειτουργικά συστήματα. Έχει ένα προεπιλεγμένο σύστημα αρχείων που είναι συμβατό με Windows (από προεπιλογή χρησιμοποιεί FAT32). Υποστηρίζει επίσης και άλλους τύπους συστημάτων αρχείων, όπως το FAT16, το NTFS, καθώς επίσης υποστηρίζει και διάφορους τύπους καρτών αποθήκευσης, όπως την MMC, την Mini, την SD και άλλες.
- Δικτύωση: Υποστηρίζει δίκτυα με πρωτόκολλο TCP/IP, καθώς και άλλες , μορφές επικοινωνίας όπως σειριακή, υπέρυθρες και Bluetooth.
- Διαχείριση Μνήμης: Η διαχείριση της μνήμης ROM είναι πολύ απλή, γιατί όλα τα αρχεία της βρίσκονται σε συγκεκριμένη μονάδα δίσκου, η οποία είναι η Z.
- Λόγω της πολυπλοκότητας του αυτό το σύστημα είναι ευάλωτο σε ιούς, trojan και άλλα κακόβουλα προγράμματα [8].

## 2.3 IOS

Το i-os είναι το λειτουργικό σύστημα της apple για smartphones. Αρχικά αναπτύχθηκε αποκλειστικά για το i-phone. Πλέον υποστηρίζει τις περισσότερες συσκευές της εταιρείας:

- i-Pod touch
- i-Pad
- Apple-TV



Εικόνα 10: Χαρακτηριστική οθόνη ios

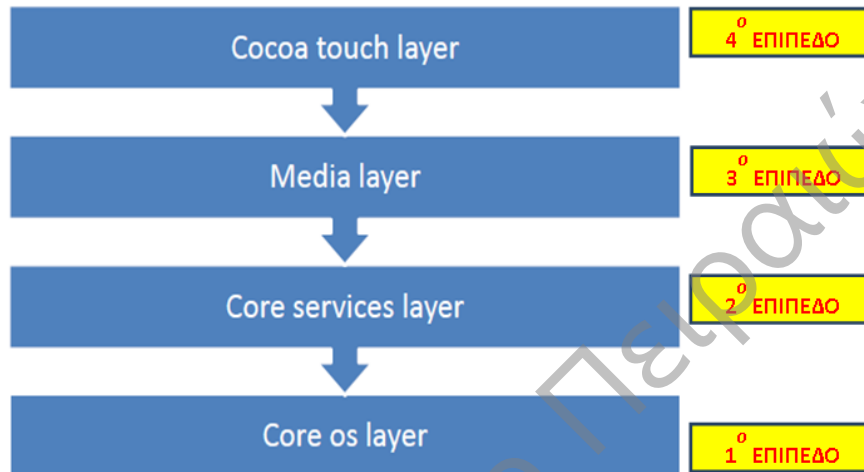
Η αρχιτεκτονική του περιλαμβάνει τα ακόλουθα επίπεδα:

- Αρχιτεκτονική του i-OS – Cocoa touch: Στο επίπεδο αυτό βρίσκεται το βασικό πλαίσιο για τη δημιουργία εφαρμογών. Είναι υπεύθυνο για τις βασικές τεχνολογίες, όπως η πολυδιεργασία, η είσοδος με αφή κ.α.
- Αρχιτεκτονική του i-OS – Media Layer: Εδώ περιέχονται οι τεχνολογίες των γραφικών, του ήχου και του βίντεο. Αυτό το επίπεδο δίνει τη δυνατότητα στους προγραμματιστές να αναπτύξουν εφαρμογές με μεγάλη ευκρίνεια ήχου και εικόνας.



- Αρχιτεκτονική του i-OS – Core Services: Περιέχει τις βασικές υπηρεσίες, που χρησιμοποιούν όλες οι εφαρμογές. Ακόμα κι αν μία εφαρμογή δεν χρησιμοποιεί αυτές τις βασικές υπηρεσίες, αρκετά τμήματά της βασίζονται σε αυτές.
- Αρχιτεκτονική του i-OS – Core OS: Αποτελείται από τα χαμηλού επιπέδου χαρακτηριστικά στα οποία στηρίζονται οι παραπάνω τεχνολογίες [7].

Σχηματικά η αρχιτεκτονική του ios φαίνεται στην ακόλουθη εικόνα.



Εικόνα 11: Αρχιτεκτονική ios

## Κεφάλαιο 3: Ανάπτυξη της Εφαρμογής

### 3.1 Εργαλεία ανάπτυξης λογισμικού

Μέσω του Android software development προκύπτουν ολοένα και νέες εφαρμογές που δημιουργούνται για το λειτουργικό σύστημα Android. Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java που χρησιμοποιεί το Android. Από τον Απρίλιο του 2011, έχουν αναπτυχθεί για το Android πάνω από 200.000 εφαρμογές, με πάνω από 3 δισεκατομμύρια downloads [16][17]. Η χρήση της πλατφόρμας Android έχει επίσης αυξηθεί, γιατί προτιμάται από τους προγραμματιστές των κινητών. Μια έρευνα τον Ιούνιο του 2011 έδειξε ότι πάνω από το 67% των κινητών χρησιμοποιούν εφαρμογές που αναπτύχθηκαν με χρήση της εν λόγω πλατφόρμας. Παρακάτω παρουσιάζονται τα πιο διαδεδομένα λογισμικά για την ανάπτυξη εφαρμογών σε Android.

- Android SDK
- Native Development Kit
- App Inventor for Android
- Hyper Next Android Creator

Στις επόμενες παραγράφους αναπτύσσεται αναλυτικά το κάθε ένα από τα παραπάνω λογισμικά.

#### 3.1.1 Android SDK

Το Android SDK παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη προγραμμάτων χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Τα χαρακτηριστικά του παρέχουν τεράστια ευελιξία και δυνατότητα ανάπτυξης έξυπνων εφαρμογών για κινητά τηλέφωνα, παρέχοντας δυνατότητες ανάπτυξης σε επιχειρηματίες όλων των κλάδων, είτε πρόκειται για αυτοματοποίηση πωλήσεων, ψυχαγωγία, παιχνίδια αλλά και κάθε άλλου είδους επιχειρήσεις. Ακόμα, παρέχει APIs για την χρήση web browser, εμφάνιση δισδιάστατων και τρισδιάστατων γραφικών, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού (ήχος, βίντεο, εικόνες), χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο.

Ένα σημαντικό γεγονός που ευνοεί την ανάπτυξη εφαρμογών είναι πως το πακέτο Android SDK συνεργάζεται με το Eclipse και συνεπώς ο προγραμματιστής μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, χωρίς να χρειάζεται να εξάγει κάθε φορά την εφαρμογή και να την εγκαθιστά σε κινητό. Επίσης, ο emulator είναι πολύ αξιόπιστος, καθώς έχει ακριβώς την ίδια συμπεριφορά με αυτή που θα είχε η εφαρμογή εάν είχε εγκατασταθεί σε ένα κινητό τηλέφωνο Android. Ένα άλλο σημαντικό πλεονέκτημα είναι το γεγονός πως σε αναβαθμίσεις του λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών

του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη- hardware- του κινητού τηλεφώνου. Παρακάτω, γίνεται ανάπτυξη του εξομοιωτή (emulator).

Το Android SDK είναι ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση των «ενδοτέρων» του Android. Παρακάτω, φαίνονται τα βασικότερα χαρακτηριστικά του.

- Εφαρμογή πλαισίου που επιτρέπει την επαναχρησιμοποίηση και την αντικατάσταση στοιχείων
- Βελτιστοποιημένη Dalvik εικονική μηχανή για κινητές συσκευές
- Ολοκληρωμένο πρόγραμμα περιήγησης, το οποίο βασίζεται στο open source WebKit
- Βελτιστοποιημένα γραφικά, τα οποία τροφοδοτούνται από μια προσαρμοσμένη βιβλιοθήκη 2D και 3D γραφικών με βάση τις προδιαγραφές του OpenGL ES 1.0 (επιτάχυνση hardware προαιρετικά)
- SQLite για δομημένη αποθήκευση δεδομένων
- Υποστήριξη πολυμέσων για αρχεία ήχου, βίντεο, ακόμα και εικόνων (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Τηλεφωνία (εξαρτώμενη από το hardware)
- Bluetooth, EDGE, 3G, WiFi (εξαρτώμενα από το hardware).
- Φωτογραφική μηχανή, GPS, πυξίδα, και επιταχυνσιόμετρο (εξαρτώμενα από το hardware)
- Πλούσιο περιβάλλον ανάπτυξης, συμπεριλαμβανομένου ενός εξομοιωτή συσκευής, εργαλεία για τον εντοπισμό σφαλμάτων, μνήμη και προφίλ απόδοσης όπως επίσης και ένα plugin για το Eclipse IDE

### 3.1.2 Ο εξομοιωτής (emulator).

Προκειμένου να γίνει ευκολότερη η διαδικασία της ανάπτυξης και αποσφαλμάτωσης μιας εφαρμογής, το Android SDK περιλαμβάνει έναν εξομοιωτή μιας εικονικής κινητής συσκευής, η οποία τρέχει το λειτουργικό του Android. Έτσι δεν είναι η αναγκαία η ύπαρξη πραγματικής κινητής συσκευής για την εκτέλεση και δοκιμή των εφαρμογών. Ο εξομοιωτής προσομοιώνει ένα μεγάλο πλήθος λειτουργιών μιας τυπικής συσκευής, η οποία τρέχει το Android:

- Παρέχει μια ποικιλία πλήκτρων πλοήγησης και ελέγχου
- Παρέχει μια οθόνη για την προβολή των εφαρμογών που τρέχουν στον εξομοιωτή
- Επιτρέπει στις εφαρμογές την χρήση των υπηρεσιών που προσφέρει η πλατφόρμα του Android, δηλαδή την κλήση άλλων εφαρμογών, την πρόσβαση στο δίκτυο, την αναπαραγωγή ήχου και βίντεο, την αποθήκευση και επαναφορά δεδομένων, την ειδοποίηση χρήστη, το γραφικό περιβάλλον του Android.

Επίσης παρέχει ένα πλήθος λειτουργιών για την ευκολότερη αποσφαλμάτωση:

- Κονσόλα για την καταγραφή της εξόδου του πυρήνα
- Προσομοίωση διακοπών (όπως η άφιξη SMS μηνύματος ή τηλεφωνικής κλήσης)
- Προσομοίωση καθυστέρησης και απώλειας στο κανάλι δεδομένων
- Προσομοίωση λήψης δεδομένων θέσης από την συσκευή GPS.

### 3.1.3 Άλλα εργαλεία του Android

Το Android SDK περιλαμβάνει μερικά ακόμη εργαλεία για την ανάπτυξη εφαρμογών:

- Το **Dalvik Debug Monitor Service (DDMS)** το οποίο επιτρέπει την διαχείριση των διεργασιών στον εξομοιωτή ή στην συσκευή. Συγκεκριμένα δίνεται η δυνατότητα port-forwarding υπηρεσιών, λήψη screenshots, εμφάνιση πληροφοριών για τον σωρό και τα νήματα, logcat, εμφάνιση πληροφοριών ράδιο και πληροφοριών διεργασιών, προσομοίωση εισερχόμενων κλήσεων και μηνυμάτων, προσομοίωση δεδομένων θέσης κ.α.
- Την **Android Debug Bridge (ADB)** η οποία επιτρέπει την διαχείριση της κατάστασης του εξομοιωτή ή της συσκευής. Μέσω του ADB είναι δυνατή η εκτέλεση εντολών φλοιού, η διαχείριση της προώθησης θυρών και η αντιγραφή από και προς την συσκευή ή τον εξομοιωτή.
- Το **Android Asset Packaging Tool (AAPT)** το οποίο δίνει την δυνατότητα δημιουργίας .apk αρχείων τα οποία περιέχουν τα εκτελέσιμα αρχεία και τους πόρους μιας εφαρμογής.
- Την **Android Interface Description Language (AIDL)** η οποία επιτρέπει την δημιουργία κώδικα που επιτρέπει σε δύο διεργασίες σε μια συσκευή βασισμένη στο Android να συνομιλούν χρησιμοποιώντας διαδιεργασιακή επικοινωνία.
- Το **SQLite3** το οποίο επιτρέπει την πρόσβαση στα δεδομένα της SQLite που δημιουργούνται από τις διάφορες εφαρμογές.
- Το **Traceview** που επιτρέπει την γραφική προβολή της ανάλυσης των trace log data που δημιουργούν οι διάφορες εφαρμογές.
- Το **mksdcard** το οποίο βοηθά στην δημιουργία εικονικού δίσκου ο οποίος μπορεί να χρησιμοποιηθεί από τον εξομοιωτή για την προσομοίωση της παρουσίας εξωτερικής αποθηκευτικής κάρτας (όπως η SD card).
- Το **dx tool** το οποίο μετατρέπει τα αρχεία .class από java bytecode σε Android bytecode.
- Το **UI/Application Exerciser Monkey** το οποίο είναι ένα πρόγραμμα που τρέχει στον εξομοιωτή και παράγει ψευδό-τυχαίες σειρές από συμβάντα

χρήστη όπως clicks, touches, gestures καθώς επίσης και έναν αριθμό από συμβάντα συστήματος.

- Το **activitycreator** το οποίο είναι ένα script που δημιουργεί Ant build αρχεία τα οποία μπορούν να χρησιμοποιηθούν για την μεταγλώττιση των εφαρμογών.

### 3.1.4 Android NDK (Native Development Kit)

Το Native Development Kit (NDK) είναι ένα σύνολο εργαλείων που επιτρέπει την ενσωμάτωση μερών (components) που κάνουν χρήση του ατόφιου κώδικα στις Android εφαρμογές. Οι Android εφαρμογές τρέχουν στην Dalvik virtual machine (εικονική μηχανή Dalvik). Είναι ένα συμπλήρωμα του SDK (Software Development Kit) παρέχει εργαλεία που επιτρέπουν την ενσωμάτωση native κώδικα στις εφαρμογές του Android. Το NDK επιτρέπει την εκτέλεση μερών των εφαρμογών, χρησιμοποιώντας γλώσσες ατόφιου κώδικα, όπως η C και η C++. Αυτό μπορεί να παρέχει πλεονεκτήματα σε συγκεκριμένες τάξεις εφαρμογών με τη μορφή της επαναχρησιμοποίησης του υπάρχοντος κώδικα και σε μερικές περιπτώσεις την αυξημένη ταχύτητα. Η ικανότητα άμεσης κλήσης της λειτουργίας στο OpenGL βελτιώνει την απόδοση γραφικών σε μεγάλες οθόνες κινητών [18].

Τα κύρια χαρακτηριστικά του Android NDK είναι τα εξής:

- Είναι ένα εργαλείο που επιτρέπει την ενσωμάτωση components που χρησιμοποιούν C και C++
- Προσφέρει παροχές σε κάποιες κλάσεις εφαρμογών
- Επαναχρησιμοποίηση ήδη υπάρχοντος κώδικα
- Αύξηση της ταχύτητας

Το Android NDK παρέχει τα εξής:

- Ένα σύνολο εργαλείων και αρχείων «χτισίματος» (build files) που χρησιμοποιούνται για να παραχθεί ατόφιος κώδικα σε βιβλιοθήκες από πηγές C και C++.
- Έναν τρόπο ενσωμάτωσης των αντίστοιχων ατόφιων βιβλιοθηκών σε ένα αρχείο πακέτου εφαρμογών (.apk), το οποίο μπορεί να αναπτυχθεί σε Android συσκευές.
- Ένα σύνολο ατόφιων επικεφαλίδων και βιβλιοθηκών συστήματος (system headers and libraries), τα οποία θα υποστηρίζονται σε όλες τις μελλοντικές εκδόσεις της Android πλατφόρμας, αρχίζοντας από το Android 1.5. Εφαρμογές που χρησιμοποιούν λειτουργίες ατόφιου κώδικα πρέπει να τρέχουν σε Android 2.3 ή μεταγενέστερα.
- Τεκμηρίωση, δείγματα και tutorials (φροντιστήρια).

Η τελευταία έκδοση του NDK υποστηρίζει τα εξής ARM σύνολα οδηγιών:

- **ARMv5TE** (που συμπεριλαμβάνει τις **Thumb-1** εντολές)
- **ARMv7-A** (που συμπεριλαμβάνει τις **Thumb-2** και **VFPv3-D16** εντολές, με προαιρετική υποστήριξη για **NEON/VFPv3-D32** εντολές)

- **x86** εντολές

Ο κώδικας μηχανής **ARMv5TE** όπως αναμένεται, τρέχει σε συσκευές Android που είναι βασισμένες στο ARM. Ο **ARMv7-A** τρέχει μόνο σε συσκευές όπως: Verizon Droid ή Google Nexus One που έχουν συμβατή CPU. Η κύρια διαφορά ανάμεσα στα δύο σύνολα εντολών είναι ότι ο ARMv7-A εμπεριέχει hardware FPU, Thumbs-2 και εντολές NEON. Μπορείτε να γίνει χρήση του ενός ή και των δύο συνόλων εντολών – το ARMv5TE είναι το default, αλλά η αλλαγή στο ARMv7-A είναι τόσο εύκολη, όσο η προσθήκη μιας και μόνο γραμμής στο αρχείο εφαρμογών Application.mk, χωρίς να χρειαστεί να αλλάξει οτιδήποτε άλλο στο αρχείο. Επίσης μπορεί να γίνει build και τις δύο αρχιτεκτονικές ταυτόχρονα και αποθήκευση όλων στο τελικό .apk αρχείο. Ολοκληρωμένες πληροφορίες παρέχονται στο CPU-ARCH-ABIS.HTML στο πακέτο NDK.

Το NDK παρέχει σταθερές επικεφαλίδες (headers) για τη libc (βιβλιοθήκη της C), libm (βιβλιοθήκη της Math), OpenGL ES (βιβλιοθήκη 3D γραφικών), τη διεπαφή JNI και άλλες βιβλιοθήκες.

Το NDK δεν θα ωφελήσει στις περισσότερες εφαρμογές. Ένας developer πρέπει να ζυγίσει τα πλεονεκτήματα έναντι των μειονεκτημάτων. Ειδικά, η χρήση ατόφιου κώδικα δεν έχει αποτέλεσμα στην αύξηση της αυτόματης απόδοσης, αλλά πάντα αυξάνει την πολυπλοκότητα της εφαρμογής. Γενικά, η χρήση ατόφιου κώδικα γίνεται μόνο αν είναι απαραίτητο για την εφαρμογή και όχι για λόγους προτιμήσεων, όπως για παράδειγμα προγραμματισμός σε C/C++.

Τυπικά καλές υποψήφιες λειτουργίες για το NDK είναι αυτές που είναι αυτοπεριοριζόμενες και που κάνουν εντατική χρήση της CPU και που παρόλα αυτά δεν δεσμεύουν πολλή μνήμη, όπως είναι η επεξεργασία σήματος, η εξομοίωση φυσικής κτλ. Η απλή επανα-κωδικοποίηση μιας μεθόδου σε C, συνήθως δεν φέρνει αποτέλεσμα μεγάλης αύξησης στην απόδοση. Κατά την εξέταση του αν πρέπει ή όχι ο προγραμματισμός πρέπει να γίνει με ατόφιο κώδικα, είναι απαραίτητη η επεξεργασία των απαιτήσεων και το αν το API του Android framework παρέχει τη λειτουργικότητα που απαιτείται.

Παρόλα αυτά το NDK μπορεί να είναι ένας αποτελεσματικός τρόπος επαναχρησιμοποίησης ενός μεγάλου μέρους του υπάρχοντος κώδικα σε C/C++.

Το **Android framework** παρέχει δυο τρόπους χρήσης του ατόφιου κώδικα:

- Συγγραφή της εφαρμογής χρησιμοποιώντας το Android framework και χρήση του **JNI** για την προσπέλαση των API που παρέχονται από το Android NDK. Αυτή η τεχνική επιτρέπει την εκμετάλλευση της ευκολίας του Android framework, αλλά επιτρέπει ακόμα και την συγγραφή ατόφιου κώδικα όταν χρειάζεται. Είναι δυνατή η εγκατάσταση εφαρμογών που χρησιμοποιούν ατόφιο κώδικα μέσω του JNI σε συσκευές που τρέχουν Android 1.5 ή μεταγενέστερο.
- Συγγραφή μιας ατόφιας μεθόδου, η οποία επιτρέπει την υλοποίηση των lifecycle callbacks με ατομικό ατόφιο κώδικα. Το Android SDK παρέχει την NativeActivity κλάση, που είναι κλάση που διευκολύνει τον προγραμματιστή, αφού ειδοποιεί τον κώδικά για κάθε δραστηριότητα

lifecycle callbacks (onCreate (), onPause (), onResume (), κτλ.). Η υλοποίηση των callbacks στον κώδικα μπορεί να γίνει με τέτοιο τρόπο που να χειρίζονται αυτά τα γεγονότα, όταν προκύπτουν. Εφαρμογές που χρησιμοποιούν ατόφιες μεθόδους πρέπει να τρέχουν σε Android 2.5 ή μεταγενέστερο.

- Δεν είναι εφικτές οι προσπελάσεις χαρακτηριστικών όπως Services και Content Providers με ατομικό κώδικα, οπότε αν η χρήση αυτών ή οποιοδήποτε άλλων API frameworks κρίνεται απαραίτητη, θα πρέπει να γίνει συγγραφή κώδικα JNI ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

Το NDK περιέχει τα API, τεκμηρίωση και ενδεικτικές εφαρμογές που βοηθούν στην συγγραφή του ατόφιου (ατομικού) κώδικα.

Το NDK περιλαμβάνει ένα σύνολο από διασταυρούμενες αλυσίδες εργαλείων (compilers, linkers κτλ.) που μπορούν να παράγουν ARM δυαδικά αρχεία σε Linux, OS X, Windows (Με Cygwin) πλατφόρμες. Περιέχει ένα σύνολο από επικεφαλίδες συστήματος για σταθερά ατόφια APIs, που είναι εγγυημένο ότι υποστηρίζονται σε όλες τις μεταγενέστερες κυκλοφορίες/εκδόσεις της πλατφόρμας:

- libc (βιβλιοθήκη C) επικεφαλίδες
- libm (βιβλιοθήκη math) επικεφαλίδες
- επικεφαλίδες JNI διεπαφής
- libz (Zlib συμπίεση) επικεφαλίδες
- liblog (Android καταγραφή) επικεφαλίδα
- OpenGL ES 1.1 και OpenGL ES 2.0 (βιβλιοθήκες 3D γραφικών) επικεφαλίδες
- Libjnigraphics (Pixel buffer access) επικεφαλίδες (for Android 2.2 και μεταγενέστερα)
- Ένα ελάχιστο σύνολο από επικεφαλίδες για υποστήριξη της C++
- OpenSL ES ατόφιες βιβλιοθήκες ήχου
- Android native application APIS

Το NDK επίσης παρέχει ένα σύστημα δόμησης (build system) που επιτρέπει την αποτελεσματική εργασία με τα source files, χωρίς να απαιτείται διαχείριση των λεπτομερειών των toolchain/πλατφόρμας/ CPU/ ABI. Δημιουργούνται πολύ μικρά build files για την περιγραφή ποιων source files θα γίνουν compile και ποια Android εφαρμογή θα τα χρησιμοποιήσει – Το Build system μεταγλωττίζει τα source files και τοποθετεί τις σκόρπιες βιβλιοθήκες κατευθείαν στο project της εφαρμογής.

**Προσοχή!** Με εξαίρεση τις παραπάνω βιβλιοθήκες, οι ατόφιες βιβλιοθήκες συστήματος στην Android πλατφόρμα δεν είναι σταθερές και μπορεί να αλλάξουν στην μελλοντική έκδοση της πλατφόρμας. Οι εφαρμογές πρέπει να κάνουν χρήση μόνο των σταθερών ατόφιων βιβλιοθηκών συστήματος που παρέχονται στο NDK.

Το πακέτο NDK περιλαμβάνει ένα σύνολο τεκμηριώσεων που περιγράφουν τις ικανότητες του NDK και πώς γίνεται η χρήση του για τη δημιουργία διαμοιρασμένων βιβλιοθηκών για τις Android εφαρμογές. Σε αυτή την έκδοση, η τεκμηρίωση παρέχεται μόνο στο NDK πακέτο που είναι προς κατέβασμα (downloadable). Συμπεριλαμβάνονται τα εξής αρχεία:

- **INSTALL.HTML** — περιγραφή του τρόπου εγκατάστασης του NDK και της ρύθμισης για το σύστημα του χρήστη
- **OVERVIEW.HTML** — παρέχει μια επισκόπηση των ικανοτήτων και της χρήσης του NDK
- **ANDROID-MK.HTML** — περιγράφει τη χρήση του αρχείου Android.mk , το οποίο ορίζει τα ατόφια source files προς μεταγλώττιση
- **APPLICATION-MK.HTML** — περιγράφει τη χρήση του αρχείου Application.mk, το οποίο περιγράφει τα ατόφια source files που απαιτούνται από την Android εφαρμογή
- **CPLUSPLUS-SUPPORT.HTML** — περιγράφει την υποστήριξη της C++ που παρέχεται στο Android NDK
- **CPU-ARCH-ABIS.HTML** — μια περιγραφή των υποστηριζόμενων αρχιτεκτονικών CPU
- **CPU-FEATURES.HTML** — μια περιγραφή της στατικής cpufeatures βιβλιοθήκης που επιτρέπει στον κώδικα της εφαρμογής να εντοπίσει την στοχευμένη οικογένεια συσκευών στην οποία ανήκει η CPU και τα προαιρετικά χαρακτηριστικά σε runtime.
- **CPU-ARM-NEON.HTML** — περιγραφή του πώς γίνεται build με προαιρετικές ARM NEON/ VFPv3-D32 εντολές.
- **CHANGES.HTML** — μια ολοκληρωμένη λίστα των αλλαγών του NDK σε όλες τις εκδόσεις.
- **DEVELOPMENT.HTML** — περιγραφή του τρόπου τροποποίησης του NDK και παραγωγή πακέτων εκδόσεων για αυτό
- **HOWTO.HTML** — πληροφορίες σχετικά με κοινές εργασίες που σχετίζονται με την ανάπτυξη του NDK
- **IMPORT-MODULE.HTML** — περιγραφή του τρόπου διαμοιρασμού και επαναχρησιμοποίησης των modules
- **LICENSES.HTML** — πληροφορίες σχετικά με τις διάφορες ανοιχτού κώδικα άδειες που διέπουν το Android NDK
- **NATIVE-ACTIVITY.HTML** — περιγραφή του τρόπου υλοποίησης των ατόφιων δραστηριοτήτων
- **NDK-BUILD.HTML** — περιγραφή της χρήσης του ndk-build script
- **NDK-GDB.HTML** — περιγραφή του τρόπου χρήσης του debugger ατόφιου κώδικα



- **PREBUILT.HTML** — πληροφορίες σχετικά με το πώς λειτουργούν οι διαμοιρασμένες και στατικές

## 3.2 Σχεδίαση της Εφαρμογής

Σκοπός της εφαρμογής είναι να παρέχει στον χρήστη της πληροφορίες για διαδρομή που πραγματοποιείται με ταξί σε πραγματικό χρόνο. Οι πληροφορίες περιλαμβάνουν την τρέχουσα θέση του και το κόστος της. Η θέση αποτυπώνεται σε χάρτη, επίσης σε πραγματικό χρόνο.

### 3.2.1 Πιθανοί Χρήστες

Πιθανοί χρήστες της εφαρμογής είναι όλοι οι χρήστες των smart phones με δυνατότητα σύνδεσης στο διαδίκτυο;.

### 3.2.2 Απαιτήσεις

#### Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις από την εφαρμογή είναι:

- Καταγραφή της τρέχουσας θέσης του χρήστη της εφαρμογής.
- Υπολογισμός του κόστους της διαδρομής με βάση την τιμολογιακή πολιτική που ακολουθείται στην Ελλάδα για τα ταξί.

#### Μη Λειτουργικές Απαιτήσεις

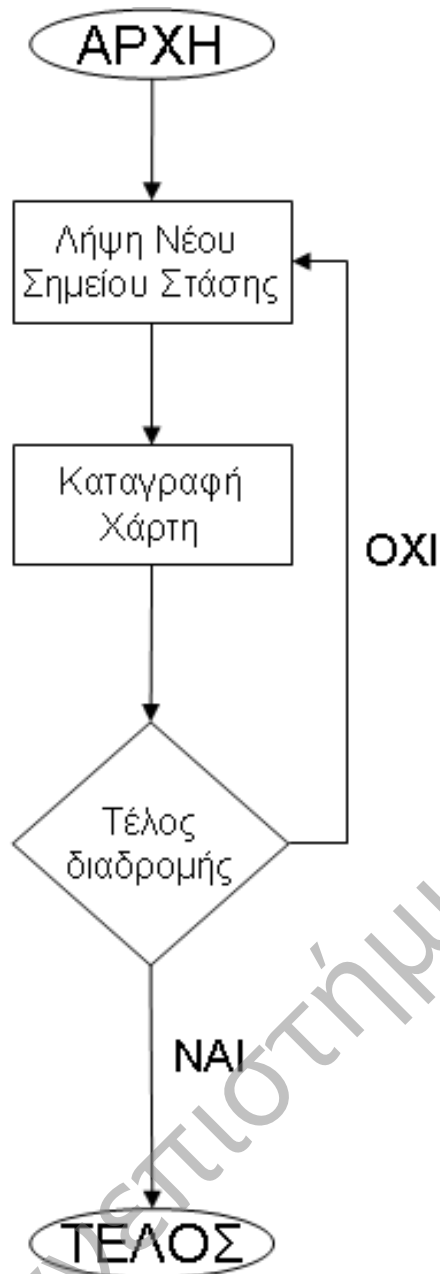
Οι μη λειτουργικές απαιτήσεις από την εφαρμογή είναι:

- Λειτουργικό Σύστημα Android
- Δυνατότητα της συσκευής για σύνδεση στο διαδίκτυο.
- Η εφαρμογή να έχει την δυνατότητα να εμφανίζει χάρτες.

### 3.2.3 Λειτουργίες

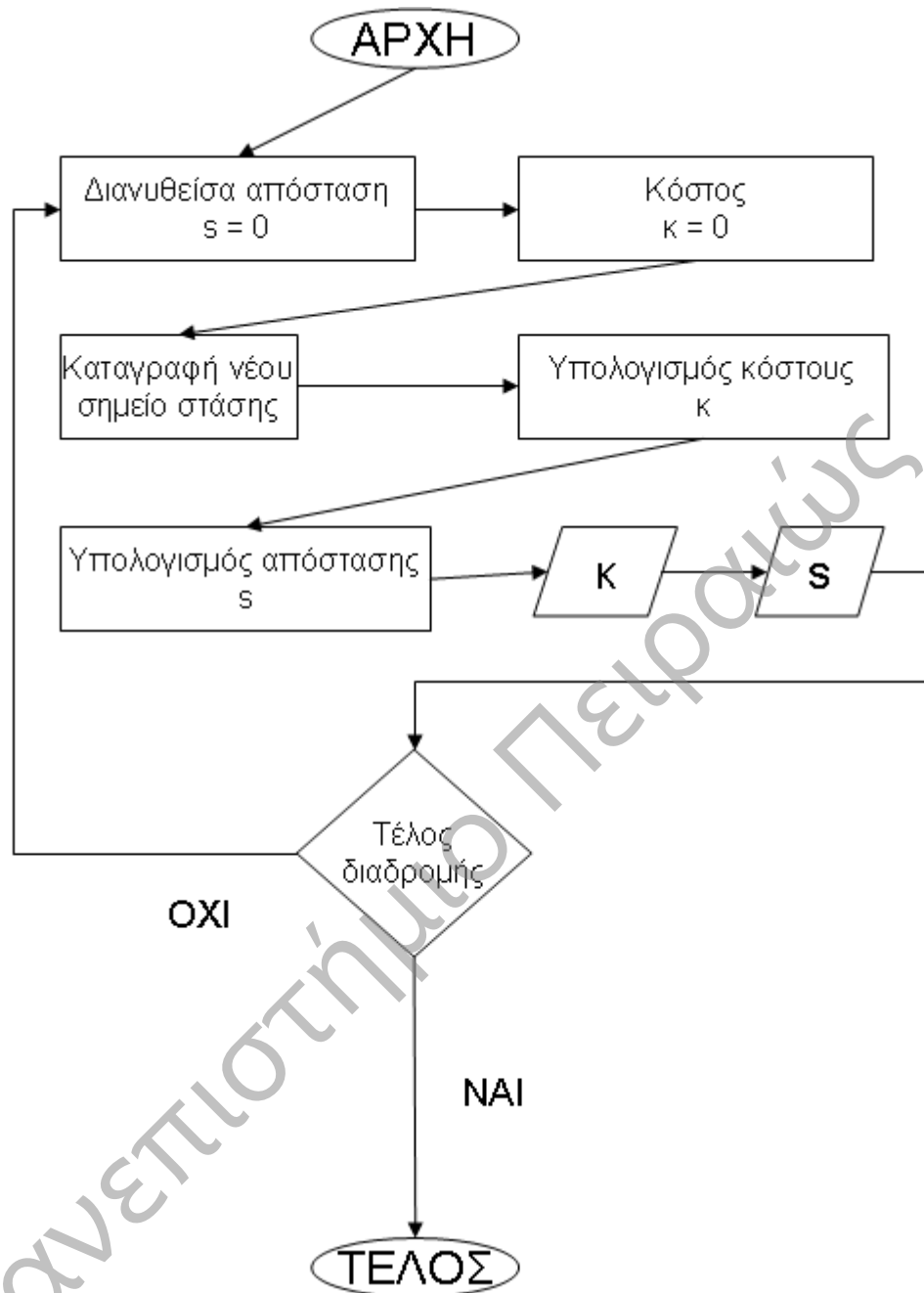
Οι λειτουργίες που απαιτείται να επιτελεί η εφαρμογή είναι:

- Καταγραφή διανυθείσας διαδρομής: Κατά την διάρκεια κίνησης του ταξί καταγράφεται σε χάρτη η διανυθείσα διαδρομή σε πραγματικό χρόνο.



Εικόνα 12: Διάγραμμα ροής για την καταγραφή της διανυθείσας διαδρομής

- Καταγραφή κόστους διανυθείσας διαδρομής: Κατά την διάρκεια κίνησης του ταξί καταγράφεται σε χάρτη το κόστος της διανυθείσας διαδρομής σε πραγματικό χρόνο.

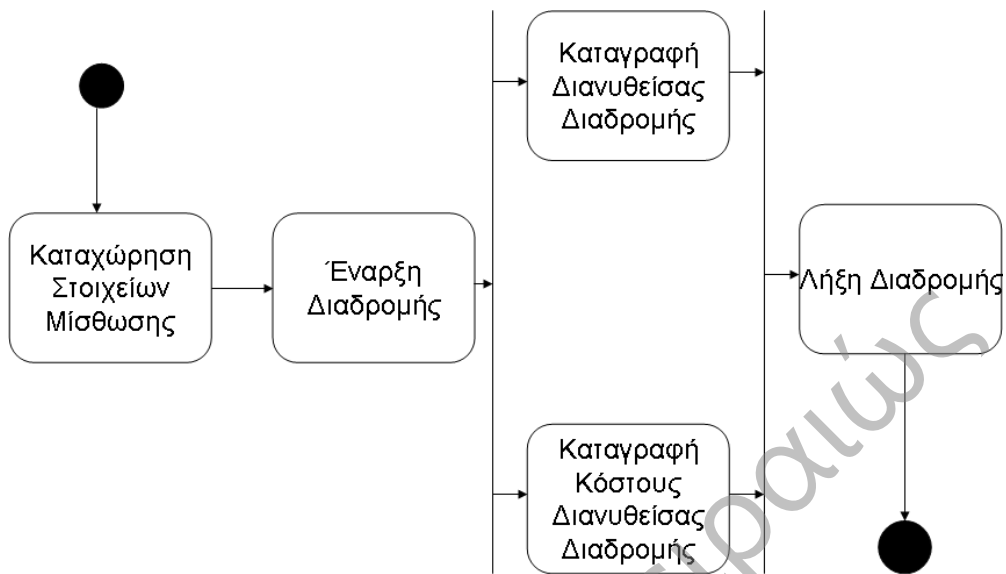


Εικόνα 13: Διάγραμμα ροής για τον υπολογισμό του κόστους διαδρομής

Συνολικά η εφαρμογή θα λειτουργεί ως εξής:

- Ο χρήστης θα καταχωρεί τα στοιχεία της μίσθωσης του ταξί που επηρεάζουν το κόστος της διαδρομής.
- Στην συνέχεια θα ορίζει την εκκίνηση της διαδρομής.
- Από την εκκίνηση της διαδρομής και μετά θα καταγράφονται η διαδρομή και το κόστος της.
- Ο χρήστης θα ορίζει το τέλος της διαδρομής και θα ολοκληρώνεται η λειτουργία της.

- Στο επόμενο διάγραμμα καταστάσεων περιγράφεται σχηματικά η διαδικασία.



Εικόνα 14: Διάγραμμα Καταστάσεων Εφαρμογής

### 3.2.4 User Interface

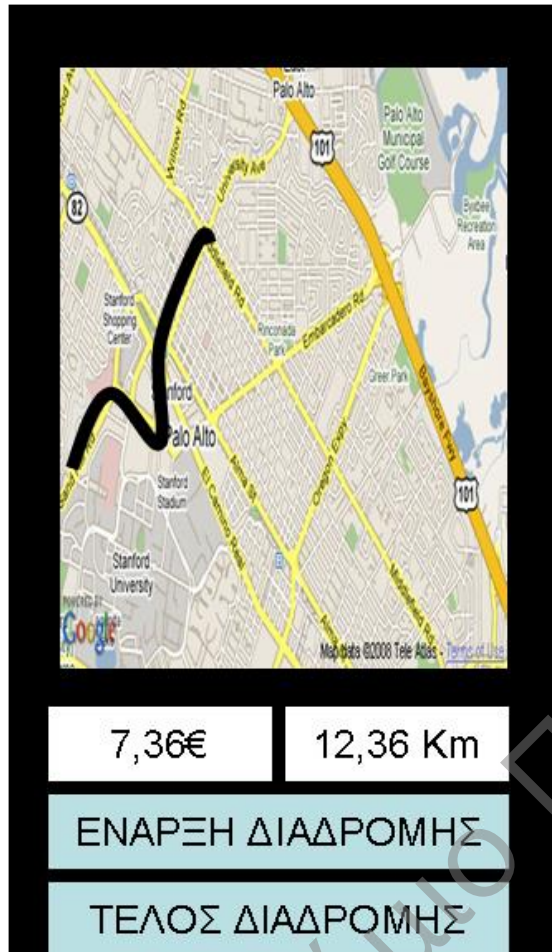
Η εφαρμογή προβλέπεται να περιλαμβάνει τις ακόλουθες οθόνες:

- Οθόνη καταγραφής στοιχείων μίσθωσης: Στην οθόνη αυτή θα υπάρχουν κατάλληλα checkboxes τα οποία θα μπορεί να τσεκάρει τα χαρακτηριστικά που επηρεάζουν το κόστος της διαδρομής. Η οθόνη θα έχει την μορφή που περιγράφεται στο επόμενο σχήμα.

	Χαρακτηριστικό 1
	Χαρακτηριστικό 1
V	Χαρακτηριστικό 1
V	Χαρακτηριστικό 1
ΚΑΤΑΧΩΡΗΣΗ	

Εικόνα 15: User Interface για την καταχώρηση των στοιχείων μίσθωσης

- Οθόνη καταγραφής διαδρομής και κόστους διαδρομής: Στην οθόνη αυτή θα υπάρχει ένας χάρτης της περιοχής που βρίσκεται ο χρήστης. Θα υπάρχει ένα πλήκτρο για μετάβαση στην οθόνη καταχώρησης στοιχείων μίσθωσης. Θα υπάρχει επίσης ένα πλήκτρο για έναρξη καταγραφής της διαδρομής και υπολογισμού του κόστους της. Επίσης θα καταγράφεται σε ένα textbox η διανυθείσα απόσταση και σε ένα άλλο textbox το κόστος της. Η μορφή της θα είναι όπως παρακάτω.



Εικόνα 16: User Interface για την λειτουργία της καταγραφής της διαδρομής και του κόστους της

## 3.3 Υλοποίηση της Εφαρμογής

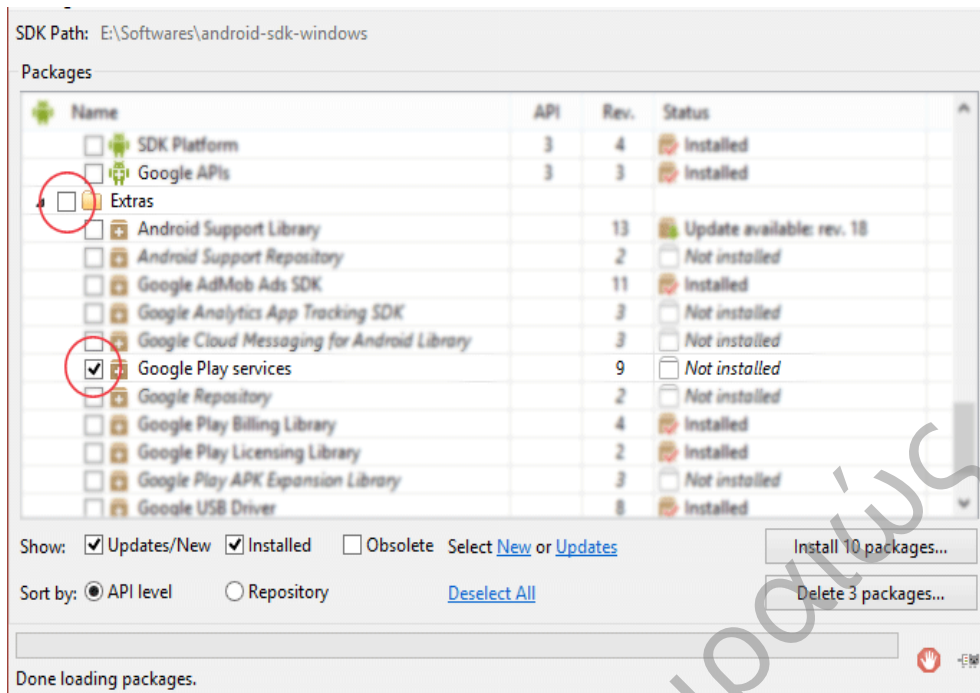
### 3.3.1 Περιβάλλον Ανάπτυξης Eclipse

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το περιβάλλον ανάπτυξης Eclipse το οποίο υπάρχει διαθέσιμο στο <http://www.eclipse.org/downloads>. Για να μπορέσει να υποστηριχθεί η ανάπτυξη android εφαρμογών χρειάζεται να γίνει εγκατάσταση του Android SDK (είναι διαθέσιμο στο <http://developer.android.com/sdk/index.html>). Το περιβάλλον ανάπτυξης προσφέρει την δυνατότητα δοκιμής των project που δημιουργούνται σε κατάλληλους εξομοιωτές. Ωστόσο οι εξομοιωτές αυτοί παρουσιάζουν δυσλειτουργίες στην εμφάνιση των χαρτών[10]. Για την δοκιμή της συγκεκριμένης εφαρμογής χρησιμοποιήθηκε συσκευή tablet Samsung.

### 3.3.2 Google Maps API

Στην εφαρμογή έχουν ενσωματωθεί οι δυνατότητες των χαρτών Google. Η διαδικασία που ακολουθήθηκε έχει ως εξής;

- Πριν γίνει οποιαδήποτε άλλη ενέργεια θα πρέπει να ληφθεί από το διαδίκτυο το Google Play Services SDK. Το Google Map API version 2 αποτελεί τμήμα αυτού. Από το βασικό μενού του Eclipse επιλέγεται:  
Eclipse → Windows → Android SDK Manager
- Στην οθόνη που εμφανίζεται τσεκάρεται η επιλογή Google Play Services κάτω από το τμήμα Extras, όπως φαίνεται στην επόμενη εικόνα



Εικόνα 17: Χρήση Google API (1)

- Στην συνέχεια θα πρέπει να εισαχθεί στο Eclipse ώστε να είναι πλέον διαθέσιμο να χρησιμοποιηθεί σαν βιβλιοθήκη. Αυτό γίνεται επιλέγοντας στο βασικό μενού του Eclipse διαδοχικά:

File → Import → Android → Existing Android Code Into Workspace

- Στην συνέχεια επιλέγεται Browse και αναζητείται το έργο Google Play Services από τον φάκελο του android sdk, όπως φαίνεται στην επόμενη εικόνα (βρίσκεται στην θέση android-sdk-windows\extras\google\google\_play\_services\libproject\google-play-services\_lib)



## Import Projects

Select a directory to search for existing Android projects



Root Directory:

Projects:

Project to Import	New Project Name	
<input checked="" type="checkbox"/> E:\Softwares\android-sdk-windo...	google-play-services_lib	<input type="button" value="Select All"/>
		<input type="button" value="Deselect All"/>
		<input type="button" value="Refresh"/>

Copy projects into workspace

Working sets

Add project to working sets

Working sets:

Εικόνα 18: Χρήση Google API (2)

- Προκειμένου να είναι εφικτή η χρησιμοποίηση των υπηρεσιών του Google Map API version , χρειάζεται ο προγραμματιστής να αποκτήσει ένα κατάλληλο API Key από την Google. Για να το αποκτήσει χρειάζεται πρώτα να δημιουργήσει το SHA1 fingerprint με την χρήση του java keytool. Αυτό επιτυγχάνεται μέσω του terminal των windows. Σε αυτό ο χρήστης μεταβαίνει στον φάκελο του jdk που είναι αποθηκευμένη η εφαρμογή keytool και εκτελεί την εντολή

```
keytool -list -v -keystore  
"%USERPROFILE%\android\debug.keystore" -alias androiddebugkey -  
storepass android -keypass android
```

Η οθόνη που αντικρίζει μετά την ενέργεια του αυτή είναι:

```

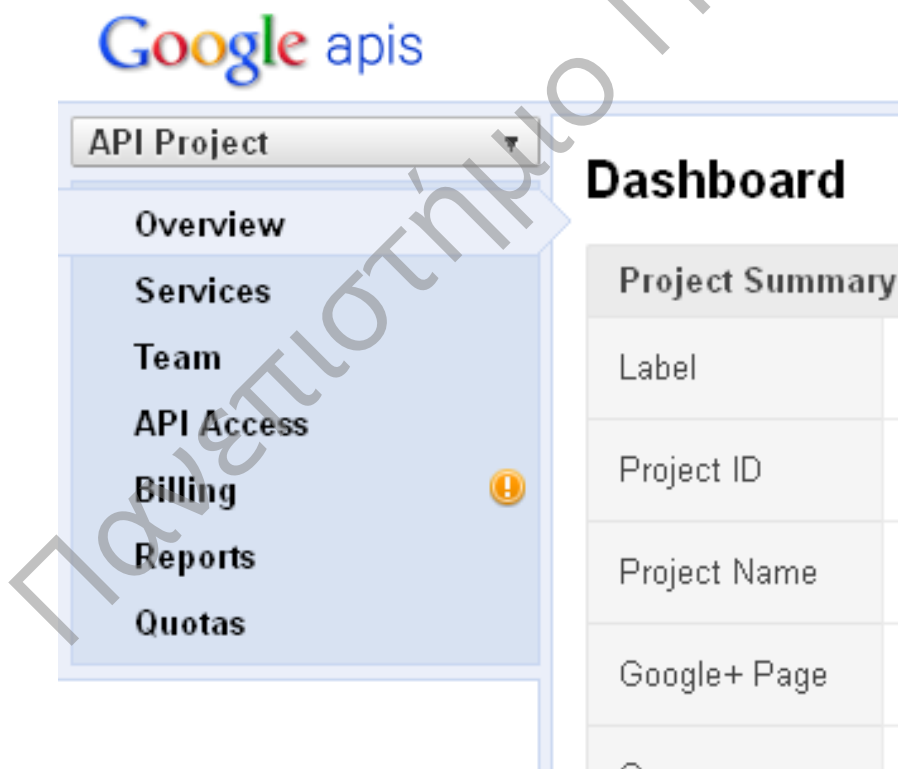
Alias name: androiddebugkey
Creation date: May 13, 2013
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 6f3eb719
Valid from: Mon May 13 02:13:08 IST 2013 until: Wed May 06 02:13:08 IST 2043
Certificate fingerprints:
  MD5: 7F:21:AF:F4:0B:67:4F:88:12:90:54:69:5E:6D:BE:DE
  SHA1: E5:0B:47:01:35:6E:77:27:D3:00:F6:54:4C:9E:8F:FF:CA:3C:60:C2
  SHA256: 3E:23:20:84:9A:77:66:52:1E:1B:E2:D5:EB:13:BE:35:FA:A8:9F:B3:86:97:F6:15:E
:13:EE:AD:F9
Signature algorithm name: SHA256withRSA
Version: 3

```

Εικόνα 19: Χρήση Google API (3)

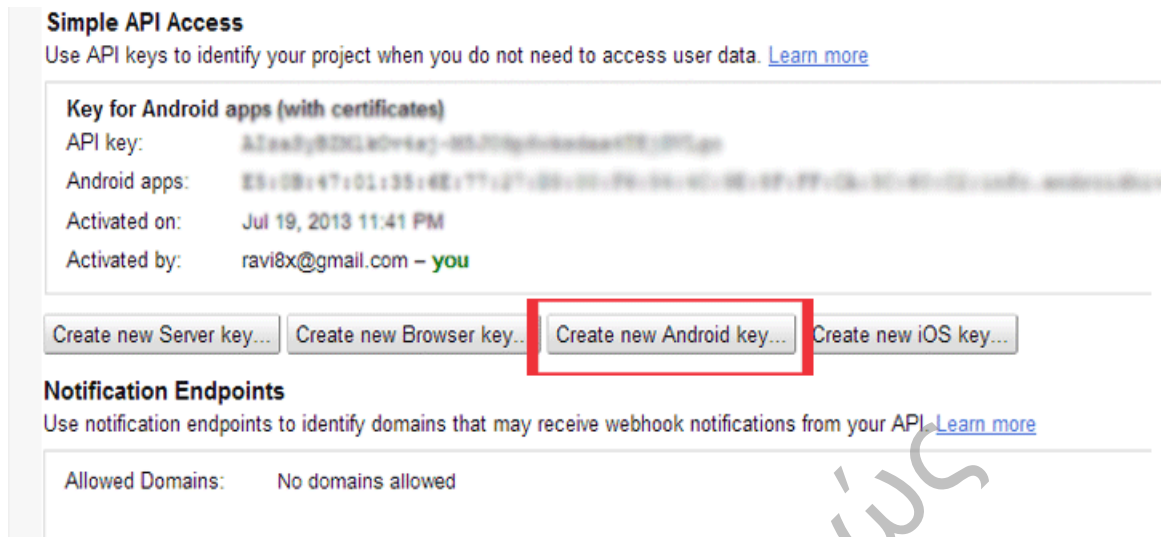
(το ζητούμενο SHA1 fingerprint είναι σημειούμενο με λευκό πλαίσιο)

- Το επόμενο βήμα είναι η μετάβαση στο κονσόλα της Google που απευθύνεται στους προγραμματιστές. Η κονσόλα αυτή βρίσκεται στο <https://code.google.com/apis/console/b/0/?noredirc>. Στην κονσόλα ο χρήστης επιλέγει services



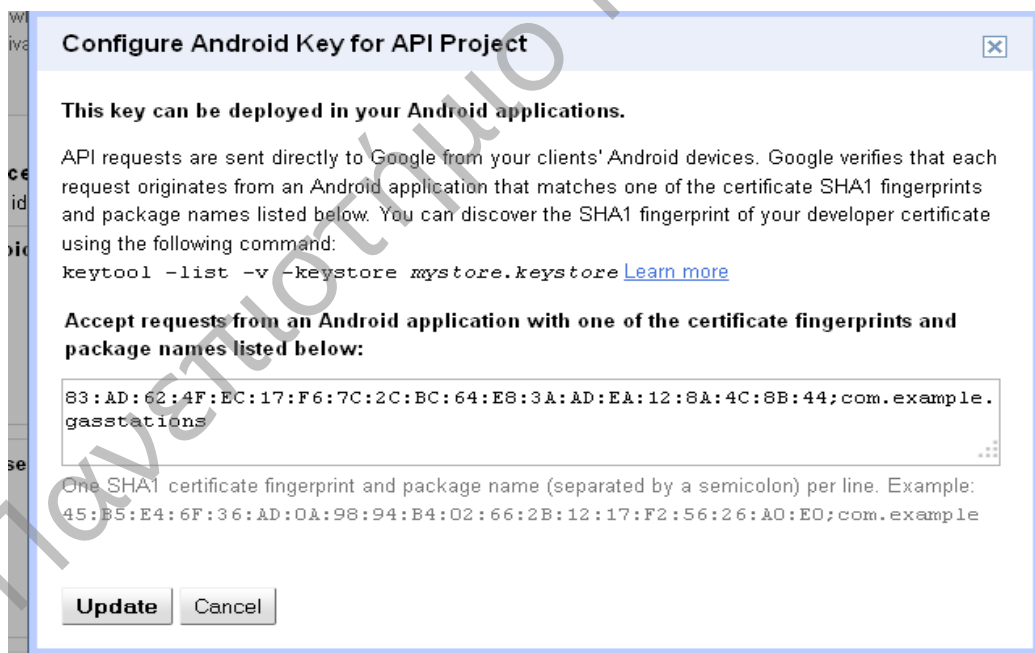
Εικόνα 20: Χρήση Google API (4)

- Έπειτα επιλέγει API access και στην οθόνη που εμφανίζεται κάνει κλικ στο πλήκτρο Generate new Android Key



Εικόνα 21: Χρήση Google API (5)

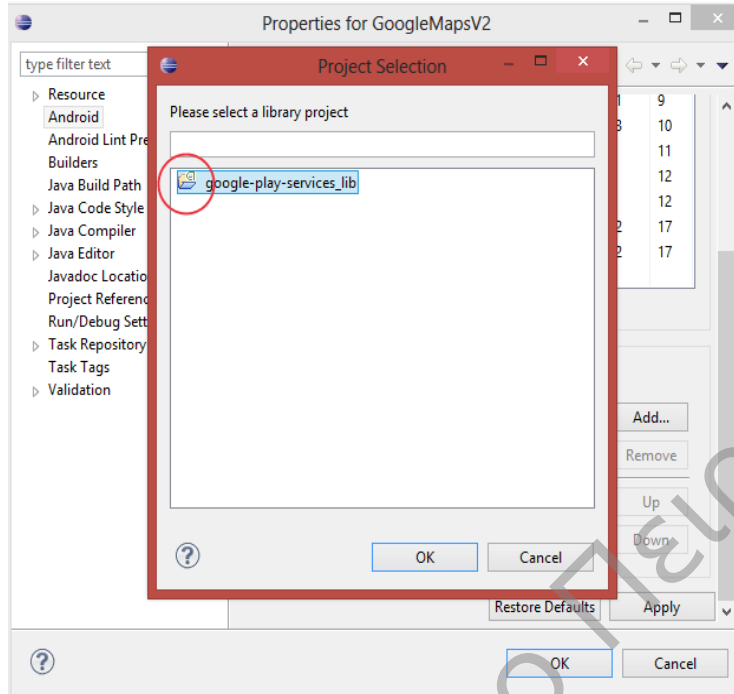
- Στο pop up παράθυρο που εμφανίζεται καταχωρείται το SHA1 fingerprint που ανακτήθηκε προηγουμένως με την εντολή keytool. Σε αυτό χρειάζεται να προσκολληθεί η ονομασία του package του project όπως φαίνεται στην επόμενη εικόνα.



Εικόνα 22: Χρήση Google API (6)

- Αφού έχουν ολοκληρωθεί η προπαρασκευαστικές ενέργειες δημιουργείται ένα έργο εφαρμογής Android. Στο έργο αυτό χρειάζεται να προστεθεί η βιβλιοθήκη του Google Play Services. Αυτό γίνεται επιλέγοντας από το βασικό μενού του Eclipse File → Properties → Android

- Στην οθόνη που εμφανίζεται γίνεται κλικ στο πλήκτρο Add και επιλέγεται το Google Play Services από το pop up παράθυρο που εμφανίζεται, όπως φαίνεται στην επόμενη εικόνα.



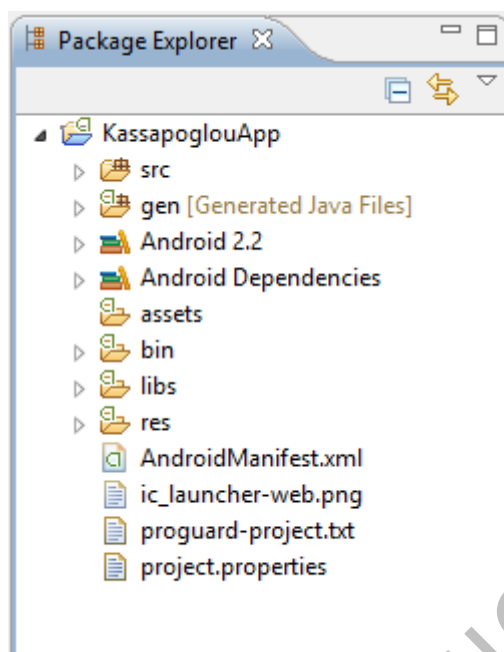
Εικόνα 23: Χρήση Google API (7)

- Στο AndroidManifest.xml καταχωρούνται βασικές πληροφορίες και ρυθμίσεις για την λειτουργία της εφαρμογής. Σε αυτό χρειάζεται να καταχωρηθούν οι ακόλουθες γραμμές που καθορίζουν τις παραμέτρους χρήσης του Google Mar API version Αυτές είναι ο προσδιορισμός του Google API Key, ο προσδιορισμός των δικαιωμάτων χρήσης.

Η προσθήκη στις διεπαφές γίνεται με αντικείμενα MapFragment και η πρόσβαση στην λειτουργικότητα τους γίνεται με χειρισμό αντικειμένων GoogleMap [10][11][12][13]

### 3.4 Κυριότερα Αρχεία

Η βασική δομή μίας εφαρμογής Android περιλαμβάνει την τους φακέλους που φαίνονται στην ακόλουθη εικόνα:



Εικόνα 24: Βασική Δομή Εφαρμογής Android (<http://osarena.net/android/aguides/mathenontas-to-android-androidmanifest-xml-is-all-about-mathimata-4-5-6.html>)

Τα κυριότερα αρχεία που περιλαμβάνονται στην δομή αυτή περιγράφονται στις επόμενες παραγράφους.

#### 3.4.1 AndroidManifest.xml

Στο αρχείο αυτό προσδιορίζονται οι παράμετροι που παίζουν ρόλο στο πως λειτουργεί η εφαρμογή. Η παρούσα εφαρμογή λειτουργεί με ενεργοποιημένη την δυνατότητα σύνδεσης στο διαδίκτυο, την χρήση των δυνατοτήτων των χαρτών της google. Παράλληλα ορίζονται το ποιες είναι οι οθόνες της εφαρμογής και το ποια θα είναι η αρχική. Ακόμα καθορίζεται το σε ποια έκδοση θα «παίξει», η ονομασία – αναγνωριστικό της τίτλος, η εικόνα που θα αποτελεί το λογότυπό της. Στο αρχείο αυτό καταγράφονται τα πακέτα java που περιλαμβάνουν την λειτουργικότητα της εφαρμογής, τα δικαιώματα που μπορεί να έχουν άλλες εφαρμογές έναντι της τρέχουσας και καθορίζει τις απαραίτητες βιβλιοθήκες που χρειάζονται για την εκτέλεση της

### 3.4.2 Activity.xml

Πρόκειται για xml αρχεία που προσδιορίζουν το πώς είναι δομημένη και το τι στοιχεία περιλαμβάνει κάθε διεπαφή. Είναι αποθηκευμένα στον φάκελο res/layout.

### 3.4.3 Strings.xml

Στο XML αρχείο αυτό καταγράφονται όλες οι συμβολοσειρές που εμφανίζονται στις διεπαφές της εφαρμογής. Οι καταχωρήσεις στο αρχείο αυτό χρησιμοποιούνται στα xml αρχεία που διαμορφώνουν τις διεπαφές της εφαρμογής με βάση το μοναδικό αναγνωριστικό που αντιστοιχεί σε κάθε συμβολοσειρά.

### 3.4.4 Activity.java

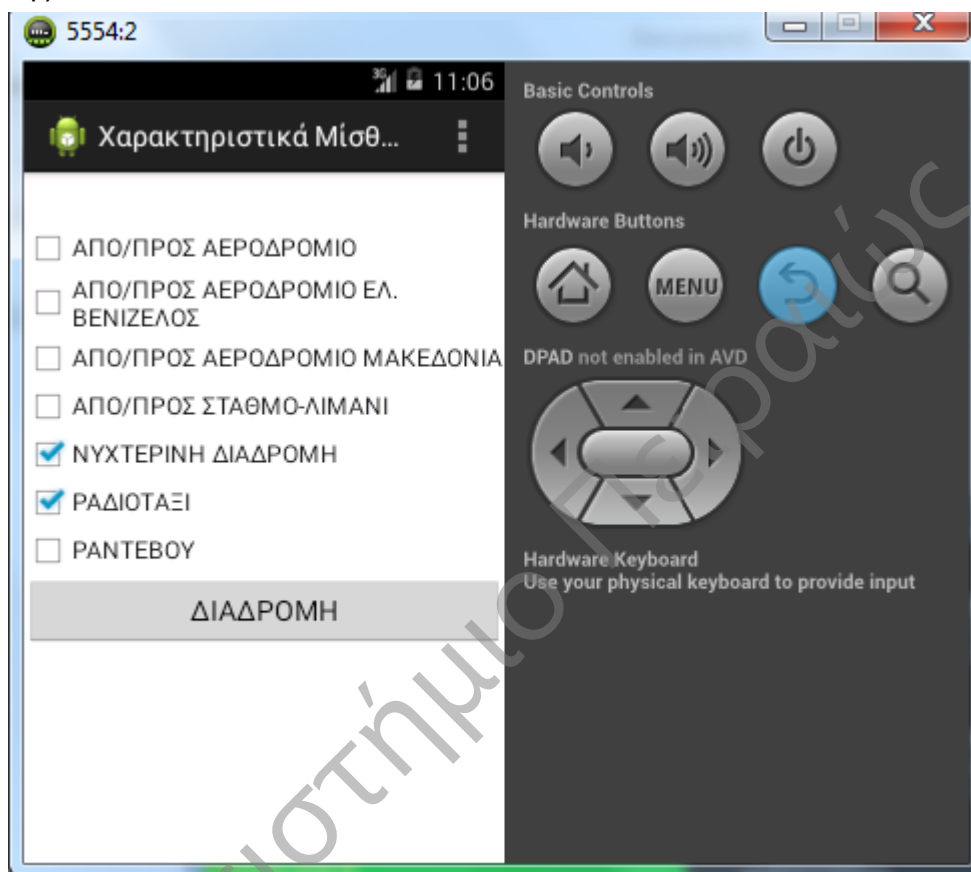
Είναι Java κλάσεις που καθορίζουν την λειτουργικότητα που αναπτύσσεται στην αντίστοιχη διεπαφή

Σε κάθε activity κλάση ορίζονται μια σειρά από events τα οποία μπορεί να είναι:

- **onCreate:** Καθορίζει την συμπεριφορά της εφαρμογής την πρώτη φορά που καλείται το activity. Αναπτύσσεται το τι συμβαίνει με την εκκίνηση της ενώ περιλαμβάνει και τους απαραίτητους listeners για τα συμβάντα που προκύπτουν κατά την εκτέλεση της εφαρμογής.
- **onStart:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το activity γίνεται ορατό.
- **onResume:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το activity χρησιμοποιείται .
- **onPause:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το τρέχον activity παγώσει και ένα προηγούμενο activity που ήταν σε αδράνεια ξεκινήσει ξανά.
- **onStop:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το activity δεν είναι ορατό.
- **onDestroy:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το activity έχει διαγραφεί από την μνήμη του συστήματος.
- **onRestart:** Καθορίζει την συμπεριφορά της εφαρμογής όταν το activity ξεκινάει ξανά.

## Κεφάλαιο 4: Χρήση της Εφαρμογής

Με την εκκίνηση της εφαρμογής ο χρήστης βλέπει την αρχική της οθόνη. Σε αυτήν μπορεί να επιλέξει είτε να καταχωρήσει τα στοιχεία μίσθωσης του ταξί, είτε να εκκινήσει την καταγραφή της διαδρομής και του κόστους της.



Εικόνα 25: Εικόνα Επιλογής Μίσθωσης (1)

Η βασική χρέωση του ταξίμετρου είναι **0.00068€/Μέτρο**. Επιλέγοντας μία από τις επιλογές αυτές το πρόγραμμα προσθέτει τις παρακάτω χρεώσεις:

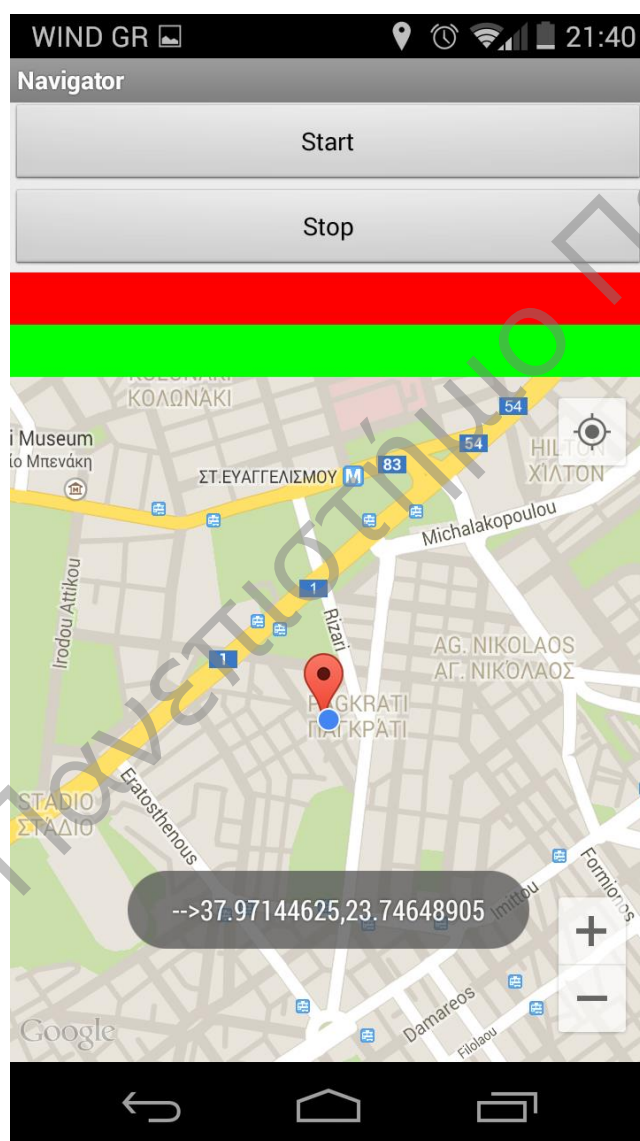
- Από ή Προς το Αεροδρόμιο: **2.30€**
- Από ή Προς το Αεροδρόμιο Ελευθέριος Βενιζέλος: **3.40€**
- Από ή Προς το Αεροδρόμιο Μακεδονία: **3.16€**
- Από ή Προς Σιδηροδρομικό Σταθμό ή Λιμάνι: **0,95€**
- Νυχτερινή Διαδρομή: **0.00119€/Μέτρο**
- Ραδιοταξί: **1.70€**
- Ραντεβού: **3.00€**

Η ελάχιστη χρέωση διαδρομής είναι **3,90€**.

Αν επιλέξει την καταγραφή της διαδρομής και του κόστους της, αρχίζουν να καταγράφονται τα σημεία από όπου περνάει το ταξί. Αυτά αποτυπώνονται στον χάρτη ενώ ταυτόχρονα φαίνεται το μέχρι εκείνη την στιγμή κόστος καθώς και η απόσταση που έχει καλυφθεί μέχρι εκείνη την ώρα.

Αν επιλέξει να καταχωρήσει τα στοιχεία της μίσθωσης τότε εμφανίζεται η αντίστοιχη οθόνη στην οποία ο χρήστης μπορεί να τσεκάρει τα αντίστοιχα checkboxes. Με την ολοκλήρωση της επιλογής μπορεί να επιλέξει να επιστρέψει στην προηγούμενη οθόνη.

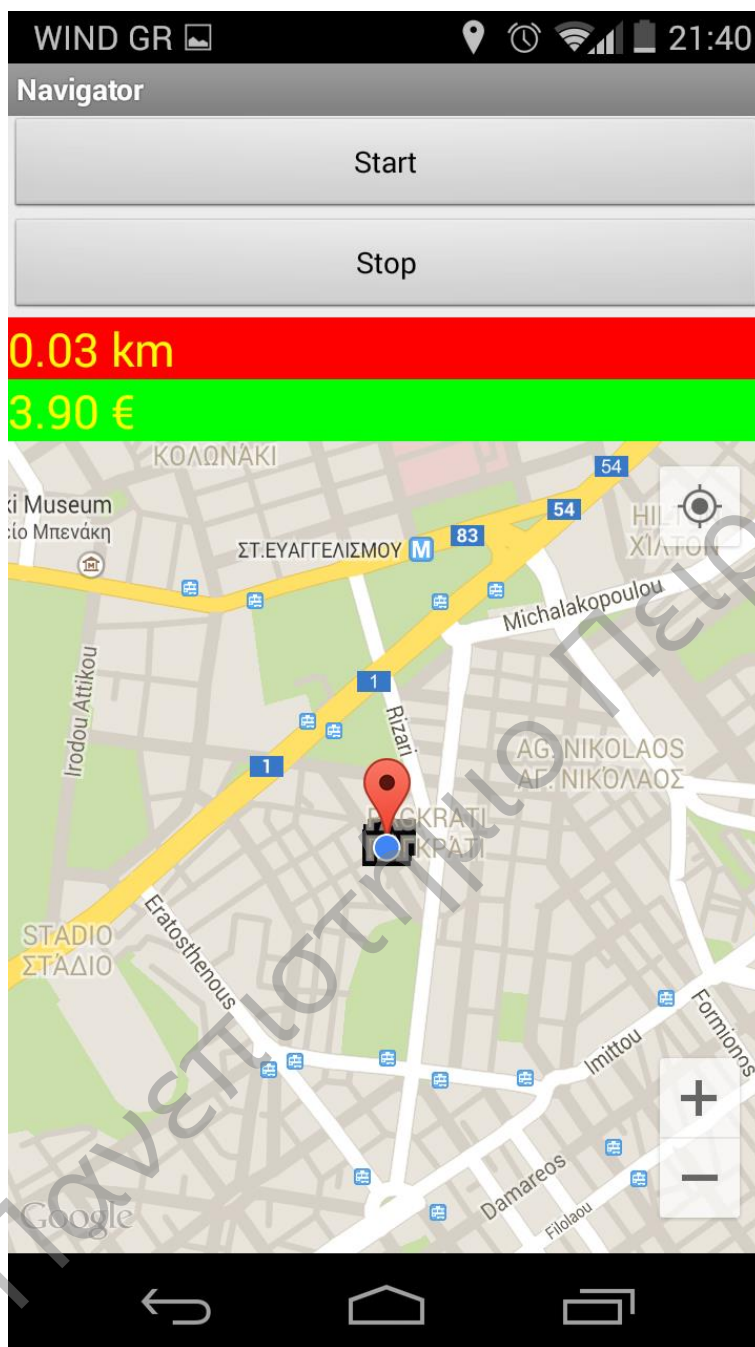
Πατώντας, λοιπόν, το κουμπί έναρξης διαδρομής η εφαρμογή λαμβάνει τις συντεταγμένες της κινητής συσκευής είτε μέσω του GPS είτε από τον Network Locator που θα αναζητήσει τις συντεταγμένες από τον πάροχο του δικτύου.



Εικόνα 26: Εμφάνιση Τωρινής Θέσης στο Χάρτη



Μετά την πρώτη οθόνη αρχίζει να καταγράφεται στον χάρτη η διαδρομή του ταξι και να ενημερώνεται η χρέωση συγχρόνως.



Εικόνα 27: Καταγραφή Διαδρομής στον χάρτη

Στο τέλος πατώντας το κουμπί Stop, σταματάει η χρέωση και εμφανίζεται το τελικό ποσό στην οθόνη. Η εφαρμογή τότε σε μεταφέρει στην οθόνη επιλογής χαρακτηριστικών μίσθωσης και γίνεται reset στον χάρτη για να είναι έτοιμος για μελλοντική χρήση.

## Επίλογος

Κύριο αντικείμενο της εργασίας ήταν η ανάπτυξη μίας εφαρμογής android. Μέσα από την διαδικασία ανάπτυξης καταδεικνύεται ότι υπάρχουν δωρεάν ισχυρά εργαλεία που διευκολύνουν την ανάπτυξη εφαρμογών υψηλών δυνατοτήτων. Το γεγονός αυτό άλλωστε οδηγεί στην ύπαρξη πληθώρα κάθε είδους εφαρμογών που διατίθεται με χαμηλό ή μηδενικό κόστος. Έτσι το λειτουργικό σύστημα Android αποκτά συγκριτικό πλεονέκτημα σε σχέση με τον ανταγωνισμό. Στην εργασία αυτή ερευνήσαμε τα διαφορετικά εργαλεία για την ανάπτυξη εφαρμογών σε Android. Μελετήθηκε η δομή και η Αρχιτεκτονική του Android αναλύοντας εκτενώς τα κυριότερα στοιχεία/αρχεία του Android. Είδαμε πότε θα πρέπει να χρησιμοποιείται το Android SDK έναντι του NDK και τις διαφορετικές προεκτάσεις τους. Τέλος αναπτύχθηκε μια εφαρμογή σε Android όπου καταγράφει και κοστολογεί την διαδρομή ενός ταξί. Η εφαρμογή περιέχει όλες τις πιθανές τιμολογήσεις που προβλέπονται ανάλογα τον χρόνο και τον τόπο αναχώρησης(Λιμάνια, Αεροδρόμια κα). Για να αναπτυχθεί η εφαρμογή χρειάστηκε να ξεπεραστούν δύο μεγάλα εμπόδια. Το πρώτο εμπόδιο είναι η συμβατότητα με τις περισσότερες αν όχι όλες τις συσκευές. Για το λόγο αυτό χρησιμοποιήθηκαν οι βιβλιοθήκες υποστήριξης (**Android Support Libraries**) του Android. Οι βιβλιοθήκες αυτές παρέχουν εκδόσεις των Android API's με backward compatibility. Δεύτερο μεγάλο εμπόδιο ήταν η υποστήριξη χάρτη σε εφαρμογή android. Για να ξεπεραστεί και το δεύτερο εμπόδιο χρησιμοποιήθηκε η τεχνολογία των **Google Play Services** και το API της Google για **Google Maps**. Με τα **Google Play Services** μπορούμε να χρησιμοποιούμε τις καινούριες υπηρεσίες της Google όπως το **Activity recognition** για την εύρεση της θέσης μας, και με το **Google Maps** για την εμφάνιση της στο χάρτη. Έτσι δημιουργούμε εφαρμογή όπου ο χρήστης μπορεί να βλέπει ανά πάσα στιγμή την θέση του στον χάρτη ακόμα και σε χώρους όπου το GPS δεν είναι διαθέσιμο.

Η εφαρμογή αυτή θα πρέπει να δοκιμαστεί εκτενώς και σε πολλές περιπτώσεις όπου το σήμα του GPS και το Activity Recognition εναλλάσσονται, ή δεν υπάρχουν καθόλου. Για επιπλέον διερεύνηση κάποιος ερευνητής θα μπορούσε να ενσωματώσει μια βάση δεδομένων για εφαρμογές όπως:

- **Εμφάνιση Εναλλακτικών διαδρομών**
- **Αξιολόγηση Διαδρομής**
- **Δημιουργία Προφίλ Χρήστη**
- **Δημιουργία Προφίλ Οδηγών Ταξί**

## Παράρτημα: Κώδικας Κυριότερων αρχείων

AndroidManifest.xml (Προσδιορισμός παραμέτρων εφαρμογής)

```
<?xml version="1.0" encoding="utf-8" ?>
- <manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="navigator.application"                android:versionCode="1"
android:versionName="1.0">
    <uses-sdk android:minSdkVersion="11" android:targetSdkVersion="17" />
    <permission
android:name="navigator.application.permission.MAPS_RECEIVE"
android:protectionLevel="signature" />
    - <!-- Required OpenGL ES 2.0. for Maps V2
-->
    <uses-feature                android:glEsVersion="0x00020000"
android:required="true" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"
/>
    - <!-- Required to show current location
-->
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.WRITE_INTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.READ_INTERNAL_STORAGE" />
```

```

    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"
/>
    <uses-permission
android:name="navigator.application.permission.MAPS_RECEIVE.permission.MAP
S_RECEIVE" />
-
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <uses-library
            android:name="com.google.android.maps"
            android:required="true" />
        <activity
            android:name="navigator.application.MainActivity"
            android:label="@string/app_name" />
        <activity
            android:name="navigator.application.NavigatorActivity"
            android:label="@string/title_activity_navigator" />
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AlzaSyAP86b6Qa29x8ETTPtpVBg8CculsiNmGv4" />
-
        <activity
            android:name="navigator.application.FeaturesActivity"
            android:label="@string/title_activity_features">
- <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    </activity>
</application>
</manifest>

```

activity\_features.xml (User Interface για την καταχώρηση των στοιχείων μίσθωσης)

```
-                                                                 <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:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".FeaturesActivity">
    <TextView                android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="@string/hello_world" />
    <CheckBox                android:id="@+id/aerodromio_all"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_alignParentTop="true" android:text="@string/aerodromio_all" />
    <CheckBox                android:id="@+id/aerodromio_venizelos"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/aerodromio_all"    android:text="@string/venizelos"
/>
    <CheckBox                android:id="@+id/aerodromio_makedonia"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/aerodromio_venizelos"
android:text="@string/makedonia" />
    <CheckBox                android:id="@+id/stahmos"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/aerodromio_makedonia"
android:text="@string/stahmos" />
    <CheckBox                android:id="@+id/nyxta"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignLeft="@+id/stahmos"    android:layout_alignParentRight="true"
android:layout_below="@+id/stahmos"    android:text="@string/nyxta" />
    <CheckBox                android:id="@+id/radiotaxi"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/nyxta"    android:text="@string/radiotaxi" />
```

```
<CheckBox                                android:id="@+id/rantevou"
android:layout_width="wrap_content"      android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/radiotaxi"    android:text="@string/rantevou" />

<Button                                  android:id="@+id/button31"
android:layout_width="wrap_content"      android:layout_height="wrap_content"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
android:layout_below="@+id/rantevou"    android:text="@string/ypologismos" />

</RelativeLayout>
```

Πανεπιστήμιο Πειραιώς

activity\_main.xml (User Interface για την καταγραφή της διαδρομής και του κόστους αυτής)

```
-                                                                 <RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:map="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"    android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity">
    <fragment android:id="@+id/map" android:layout_width="fill_parent"
android:layout_height="match_parent"    android:layout_above="@+id/stopbtn"
android:layout_alignParentLeft="true"    android:layout_alignParentRight="true"
class="com.google.android.gms.maps.SupportMapFragment"
android:apiKey="AlzaSyAP86b6Qa29x8ETTPtVBg8CcusiNmGv4"
android:clickable="true" android:debuggable="true" android:enabled="true" />
    <TextView                                android:id="@+id/timeelapsedlabel"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentBottom="true" android:layout_alignParentLeft="true"
android:text="@string/time"
android:textAppearance="?android:attr/textAppearanceSmall" />
    <Chronometer                                android:id="@+id/chronometer1"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_alignTop="@+id/timeelapsedlabel"
android:layout_toRightOf="@+id/timeelapsedlabel"
android:text="@string/empty" />
    <TextView                                android:id="@+id/velocitytext"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_above="@+id/timeelapsedlabel"
android:layout_alignParentLeft="true"    android:text="@string/velocity"
android:textAppearance="?android:attr/textAppearanceSmall" />
    <TextView                                android:id="@+id/distancetext"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_above="@+id/velocitytext" android:layout_alignParentLeft="true"
android:text="@string/distance"
android:textAppearance="?android:attr/textAppearanceSmall" />
    <TextView                                android:id="@+id/velocityvalue"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
```

```

android:layout_above="@+id/chronometer1"
android:layout_toRightOf="@+id/velocitytext"    android:text="@string/empty"
android:textAppearance="?android:attr/textAppearanceSmall" />
    <TextView                                android:id="@+id/distancevalue"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_above="@+id/velocityvalue"
android:layout_toRightOf="@+id/distancetext"    android:text="@string/empty"
android:textAppearance="?android:attr/textAppearanceSmall" />
    <Button                                android:id="@+id/startbtn"
android:layout_width="wrap_content"    android:layout_height="wrap_content"
android:layout_above="@+id/distancetext" android:layout_alignParentLeft="true"
android:text="@string/start" />
    <Button                                android:id="@+id/stopbtn"
style="?android:attr/buttonStyleSmall"    android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_alignTop="@+id/startbtn"
android:layout_toRightOf="@+id/startbtn" android:text="@string/stop" />
</RelativeLayout>

```



MainActivity.java (Περιγράφει την λειτουργία της καταγραφής διαδρομής και κόστους αυτής)

```
package navigator.application;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import com.google.android.gms.location.LocationClient;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.GoogleMap.OnMyLocationChangeListener;
import com.google.android.gms.maps.MapInitializer;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.GroundOverlayOptions;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.drawable.Drawable;
import android.support.v4.app.FragmentActivity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
```

```

import android.widget.Chronometer;
import android.widget.TextView;
import android.widget.Toast;
import android.os.SystemClock;

import com.google.android.gms.common.ConnectionResult;
import
com.google.android.gms.common.GooglePlayServicesNotAvailableException;
import com.google.android.gms.common.GooglePlayServicesUtil;

import com.google.android.gms.maps.SupportMapFragment;

public class MainActivity extends FragmentActivity implements
OnMyLocationChangeListener {
    public String extras;
    protected LocationManager locationManager;
    protected LocationListener locationListener;
    protected Context context;
    //ΛΙΣΤΑ ΣΗΜΕΙΩΝ ΚΑΙ ΠΕΡΙΓΡΑΦΩΝ ΑΥΤΩΝ
    ArrayList<String> Descr;
    ArrayList<LatLng> Ps;

    //1: ΔΙΑΔΙΚΑΣΙΑ ΚΑΤΑΓΡΑΦΗΣ ΔΙΑΔΡΟΜΗΣ 0: ΔΕΝ ΕΙΝΑΙ ΣΕ
ΔΙΑΔΙΚΑΣΙΑ ΚΑΤΑΓΡΑΦΗΣ ΔΙΑΔΡΟΜΗΣ
    int RecordStatus = 0;

    String Filename;
    String lat;
    String provider;

    Location CurrentLocation;
    double CurrentDistance;
    double X,Y;
    protected double latitude;
    protected double longitude;

```

```

        protected boolean gps_enabled,network_enabled;
        Drawable drawable;
        //Google Map
        private GoogleMap map;
    TextView distance;
    TextView velocity;
    TextView pointdescription;
    Chronometer chronometer;
    ArrayList<LandMark> landmarks;
    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Intent myIntent = getIntent(); // gets the previously created
intent
        extras = myIntent.getStringExtra("extras");
        Descr = new ArrayList<String>();
        Ps = new ArrayList<LatLng>();

        // ΔΙΑΘΕΣΙΜΟΤΗΤΑ ΤΟΥ GOOGLEPLAY
        int status =
        GooglePlayServicesUtil.isGooglePlayServicesAvailable(getBaseContext());

        //ΕΜΦΑΝΙΣΗ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΤΟΥ
        if(status!=ConnectionResult.SUCCESS){ // ΔΕΝ ΥΠΑΡΧΕΙ
ΔΙΑΘΕΣΙΜΟΤΗΤΑ
            int requestCode = 10;
            Dialog dialog = GooglePlayServicesUtil.getErrorDialog(status, this,
requestCode);
            dialog.show();

        }
        else { //ΔΙΑΘΕΣΙΜΟ
            SupportMapFragment fm = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);

```

```

// ΛΗΨΗ ΤΟΥ ΧΑΡΤΗ
map = fm.getMap();
// ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΩΝ ΤΟΠΟΘΕΣΙΩΝ ΤΟΥ ΧΑΡΤΗ
map.setMyLocationEnabled(true);
// ΕΝΕΡΓΟΠΟΙΕΙΤΑΙ Ο LISTENER ΓΙΑ ΤΗΝ ΑΛΛΑΓΗ ΤΟΠΟΘΕΣΙΑΣ
map.setOnMyLocationChangeListener(this);
}

if (map != null){
//ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΤΡΕΧΟΝΤΟΣ ΣΗΜΕΙΟΥ
LatLng latLng = new LatLng(latitude, longitude);
//ΕΜΦΑΝΙΣΗ ΤΟΥ ΤΡΕΧΟΝΤΟΣ ΣΗΜΕΙΟΥ ΣΤΟΝ ΧΑΡΤΗ
map.moveCamera(CameraUpdateFactory.newLatLng(latLng));
//ΖΟΟΜ ΣΤΟΝ ΧΑΡΤΗ
map.animateCamera(CameraUpdateFactory.zoomTo(15));
//ΕΜΦΑΝΙΣΗ ΣΤΙΓΜΑΤΟΣ
        GroundOverlayOptions newarkMap = new
GroundOverlayOptions()
        .image(BitmapDescriptorFactory.fromResource(R.drawable.pointer))
        .position(latLng, 86f, 65f);
        map.addGroundOverlay(newarkMap);
}

landmarks = new ArrayList<LandMark>();
final Button startbtn = (Button) findViewById(R.id.startbtn);
final Button stopbtn = (Button) findViewById(R.id.stopbtn);

        distance = (TextView) findViewById(R.id.distancevalue);
        velocity = (TextView) findViewById(R.id.velocityvalue);
        chronometer = (Chronometer)
findViewById(R.id.chronometer1);
        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

try {

```

```

        Process proc = Runtime.getRuntime().exec( "chmod
0777 thehistoryfile" );
    } catch (IOException e2) {
        // TODO Auto-generated catch block
        e2.printStackTrace();
    }

    X = latitude;
    Y = longitude;

    //Ο ΧΡΗΣΤΗΣ ΚΑΝΕΙ ΚΛΙΚ ΣΤΟ START
    startbtn.setOnClickListener(new Button.OnClickListener() {
        public void onClick(View v) {
            //ΑΡΧΙΖΕΙ ΤΟ ΧΡΟΝΟΜΕΤΡΟ

            chronometer.setBase(SystemClock.elapsedRealtime());
            chronometer.start();
            //ΑΡΧΙΚΟΠΟΙΗΣΗ ΑΠΟΣΤΑΣΗΣ
            CurrentDistance = 0;
            //ΚΑΤΑΣΤΑΣΗ ΕΓΓΡΑΦΗΣ
            RecordStatus = 1;
        }
    });

    //Ο ΧΡΗΣΤΗΣ ΚΑΝΕΙ ΚΛΙΚ ΣΤΟ STOP
    stopbtn.setOnClickListener(new Button.OnClickListener() {
        public void onClick(View v) {
            //ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΕΝΑ ΠΡΟΣΩΡΙΝΟ ΑΡΧΕΙΟ ΣΤΟ
            ΟΠΟΙΟ ΚΑΤΑΓΡΑΦΟΝΤΑΙ ΤΑ
            //ΣΤΟΙΧΕΙΑ ΤΗΣ ΔΙΑΔΡΟΜΗΣ
            //ΤΟ ΧΡΟΝΟΜΑΤΡΟ ΣΤΑΜΑΤΑΕΙ
            chronometer.stop();
            //ΒΓΑΙΝΟΥΜΕ ΑΠΟ ΤΗΝ ΚΑΤΑΣΤΑΣΗ ΕΓΓΡΑΦΗΣ
            RecordStatus = 0;
        }
    });

```

```

    }

    @Override
    protected void onResume() {
        super.onResume();
    }

    //ΥΠΟΛΟΓΙΖΕΙ ΚΑΙ ΕΠΙΣΤΡΕΦΕΙ ΤΗΝ ΑΠΟΣΤΑΣΗ ΜΕΤΑΞΥ ΔΥΟ ΣΗΜΕΙΩΝ
    ΣΤΟΝ ΧΑΡΤΗ
    @SuppressWarnings("null")
    public double distance(double x,double x1,double y1,double
x2,double y2){
        Location l1 = new Location("l1");
        Location l2 = new Location("l2");
        l1.setLatitude(x1);
        l1.setLongitude(y1);
        l2.setLatitude(x2);
        l2.setLongitude(y2);
        return x + l1.distanceTo(l2);
    }

    protected boolean isRouteDisplayed() {
        // TODO Auto-generated method stub
        return false;
    }

    //ΟΤΑΝ ΑΛΛΑΖΕΙ Η ΤΟΠΟΘΕΣΙΑ ΤΗΣ ΣΥΣΚΕΥΗΣ
    @Override
    public void onMyLocationChange(Location location) {
        double Xpre;
        double Ypre;
        Xpre = X;
        Ypre = Y;
        X = location.getLatitude();
        Y = location.getLongitude();
        CurrentLocation = location;
    }

```

```

        //ΛΑΜΒΑΝΟΝΤΑΙ ΟΙ ΣΥΝ-ΝΕΣ ΤΟΥ ΣΗΜΕΙΟΥ ΚΑΙ
        ΚΑΤΑΧΩΡΟΥΝΤΑΙ ΣΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΚΛΑΣΗΣ
        latitude = location.getLatitude();
        longitude = location.getLongitude();
        //ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΤΡΕΧΟΝΤΟΣ ΣΗΜΕΙΟΥ
        LatLng latLng = new LatLng(latitude, longitude);
        //ΕΣΤΙΑΖΕΤΑΙ ΤΟ ΣΗΜΕΙΟ
        map.moveCamera(CameraUpdateFactory.newLatLng(latLng));
        //ΓΙΝΕΤΑΙ ZOOM ΣΤΟ ΤΡΕΧΟΝ ΣΗΜΕΙΟ
        map.animateCamera(CameraUpdateFactory.zoomTo(15));
        //ΑΝ Η ΕΦΑΡΜΟΓΗ ΒΡΙΣΚΕΤΑΙ ΣΕ ΦΑΣΗ ΚΑΤΑΓΡΑΦΗΣ

        if (RecordStatus == 1){
            //ΕΜΦΑΝΙΣΗ ΣΤΙΓΜΑΤΟΣ
            GroundOverlayOptions newarkMap = new
            GroundOverlayOptions()
            .image(BitmapDescriptorFactory.fromResource(R.drawable.pointer))
            .position(latLng, 86f, 65f);
            map.addGroundOverlay(newarkMap);
            //ΚΑΤΑΓΡΑΦΗ ΤΗΣ ΜΕΧΡΙ ΤΩΡΑ ΔΙΑΔΡΟΜΗΣ
            PolylineOptions rectOptions = new
            PolylineOptions().add(latLng); // Closes the polyline.
            Polyline polyline = map.addPolyline(rectOptions);

            //ΚΑΤΑΓΡΑΦΗ ΣΤΗΝ ΛΙΣΤΑ ΤΩΝ ΣΗΜΕΙΩΝ
            Ps.add(latLng);
            Descr.add(" ");
            //ΥΠΟΛΟΓΙΣΜΟΣ ΔΙΑΝΥΘΗΣΑΣ ΑΠΟΣΤΑΣΗΣ
            if ((X==--1000) || (Y==--1000) || (X!=0) || (Y!=0)){
                X = location.getLatitude();
                Y = location.getLongitude();
            }

            CurrentDistance =
            distance(CurrentDistance,Xpre,Ypre,location.getLatitude(),location.getLongitude())
            ;

```

```

//ΟΙ ΝΕΕΣ ΤΡΕΧΟΥΣΕΣ ΣΥΝΤΕΤΑΓΜΕΝΕΣ
X = location.getLatitude();
Y = location.getLongitude();
//ΕΜΦΑΝΙΣΗ ΤΗΣ ΑΠΟΣΤΑΣΗΣ ΠΟΥ ΕΧΕΙ ΔΙΑΝΥΘΕΙ
distance.setText(String.format("%.2f",CurrentDistance/1000)+
km");

long timeElapsed = SystemClock.elapsedRealtime() -
chronometer.getBase();
//ΥΠΟΛΟΓΙΣΜΟΣ ΤΗΣ ΜΕΣΗΣ ΤΑΧΥΤΗΤΑΣ
double cost = getCurrentCost(CurrentDistance,extras);
//double u = CurrentDistance / (timeElapsed/1000);
//ΕΜΦΑΝΙΣΗ ΤΗΣ ΜΕΣΗΣ ΤΑΧΥΤΗΤΑΣ ΣΤΗΝ ΟΘΟΝΗ
velocity.setText(String.format("%.2f",cost) + " € ");
}
}

double getCurrentCost(double distance, String extras){

    double cost = 0;

    cost = 1.19; //ΑΠΟ - ΠΡΟΣ ΤΟ ΑΕΡΟΔΡΟΜΙΟ
    if (extras.contains("aerodromio_all")){
        cost = cost + 2.30;
    }
    //ΑΠΟ - ΠΡΟΣ ΤΟ ΑΕΡΟΔΡΟΜΙΟ ΕΛ ΒΕΝΙΖΕΛΟΣ
    if (extras.contains("aerodromio_venizelos")){
        cost = cost + 3.40;
    }
    //ΑΠΟ - ΠΡΟΣ ΤΟ ΑΕΡΟΔΡΟΜΙΟ ΜΑΚΕΔΟΝΙΑ
    if (extras.contains("aerodromio_makedonia")){
        cost = cost + 3.16;
    }
    //ΑΠΟ - ΠΡΟΣ ΣΤΑΘΜΟ - ΛΙΜΑΝΙ
    if (extras.contains("stahmos")){

```



```

        cost+= 0.95;
    }
    //RADIO TAXI
    if (extras.contains("radiotaxi")){
        cost = cost + 1.70;
    }
    //RANTEBOY
    if (extras.contains("rantevou")){
        cost = cost + 3;
    }
    //APOSKEVES

    double costpermeter = 0.068/1000;
    //YPOLOGISMOS APOSTASHS APO GOOGLE
    //distance
    meters(nomos_input1,dhmos_input1,address_input1,nomos_input2,dhmos_input
2,address_input2);
    //AN EGINE O YPOLOGISMOS
    if (distance>0){
    if ((distance > 24000) || (extras.contains("nyxta"))){
        costpermeter = 1.19/1000;
    }
    cost += costpermeter*distance;
    //ELAXISTO KOSTOS
    if (cost < 3.90){
        cost = 3.90;
    }
    String sd = String.format( "%6.2f", cost );
    cost = Double.parseDouble( sd );
    }
return cost;
}

public boolean onCreateOptionsMenu(Menu menu){
    super.onCreateOptionsMenu(menu);
    MenuInflater oMenu = getMenuInflater();

```

```
oMenu.inflate(R.menu.main, menu);
return true;
}

public boolean onOptionsItemSelected(MenuItem item){
    switch(item.getItemId()){
        case R.id.mapStreet:
            map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
            return true;

        case R.id.mapSat:
            map.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
            return true;
    }
    return false;
}
}
```

FeaturesActivity.java (Περιγράφει την λειτουργία καταχώρησης των στοιχείων μίσθωσης)

```
package navigator.application;
import android.os.Bundle;
import android.os.StrictMode;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.TextView;
import android.widget.Toast;

public class FeaturesActivity extends Activity {
    public String extras;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_features);

        final Button go = (Button) findViewById(R.id.button31);
        final Button back = (Button) findViewById(R.id.prev);
        final CheckBox aerodromio_all = (CheckBox)
findViewById(R.id.aerodromio_all);
        final CheckBox aerodromio_venizelos = (CheckBox)
findViewById(R.id.aerodromio_venizelos);
        final CheckBox aerodromio_makedonia = (CheckBox)
findViewById(R.id.aerodromio_makedonia);
        final CheckBox nyxta = (CheckBox) findViewById(R.id.nyxta);
        final CheckBox radiotaxi = (CheckBox)
findViewById(R.id.radiotaxi);
        final CheckBox rantevou = (CheckBox)
findViewById(R.id.rantevou);
        final CheckBox stahmos = (CheckBox)
findViewById(R.id.stahmos);
```

```

        final TextView aposkeves = (TextView)
findViewById(R.id.aposkeves_text);
        Intent myIntent = getIntent(); // gets the previously created
intent

        //EPITREPEI NA GINEI SYNDESH STO INTERNET APO TO
VASIKO THREAD

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        //ME TO PLHKTRO EPOMENO GINETAI O IPOLOGISMOS TOY
KOSTOYS

        go.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                extras = "-";
                //APO - PROS TO AERODROMIO
                if (aerodromio_all.isChecked()){
                    extras = ",aerodromio_all";
                }
                //APO - PROS TO AERODROMIO EL VENIZELOS
                if (aerodromio_venizelos.isChecked()){
                    extras = ",aerodromio_venizelos";
                }
                //APO - PROS TO AERODROMIO MAKEDONIA
                if (aerodromio_makedonia.isChecked()){
                    extras = ",aerodromio_makedonia";
                }
                //APO - PROS STATHMO - LIMANI
                if (stahmos.isChecked()){
                    extras = ",stahmos";
                }
                //RADIO TAXI
                if (radiotaxi.isChecked()){
                    extras = ",radiotaxi";
                }
            }
        });

```

```

        //RANTEBOY
        if (rantevou.isChecked()){
            extras = ",rantevou";
        }
        //APOSKEVES
        if (aposkeves.getText().toString().isEmpty()){
            extras = ",valitses-0";
        }
        else {
            extras = ",valitses-" +
aposkeves.getText().toString();
        }
        if (nyxta.isChecked()){
            extras = ",nyxta";
        }
        //KALEITAI H TELIKH OTHONI
        Intent intent = new Intent(FeaturesActivity.this,
MainActivity.class);
        intent.putExtra("extras", extras);
        startActivity(intent);
    }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.features, menu);
    return true;
}
}
}

```

LandMark.java (Περιγράφει κλάση αντικειμένων για την απεικόνιση σημείων στον χάρτη)

```
package navigator.application;

//ΚΛΑΣΗ ΠΟΥ ΠΑΡΙΣΤΑΝΕΙ ΕΝΑ ΣΗΜΕΙΟ ΠΑΝΩ ΣΤΟΝ ΧΑΡΤΗ
public class LandMark {
    //ΣΥΝΤΕΤΑΓΜΕΝΕΣ
    double lan;
    double lon;
    //ΠΕΡΙΓΡΑΦΗ ΣΗΜΕΙΟΥ
    String description;

    //CONSTRUCTOR
    public void LandMark(){
        lan = 0;
        lon = 0;
        description = "";
    }

    //ΕΙΣΑΓΩΓΗ ΣΥΝΤΑΤΑΓΜΕΝΩΝ
    public void setLan(double x){
        lan = x;
    }

    public void setLon(double x){
        lon = x;
    }

    //ΕΙΣΑΓΩΓΗ ΠΕΡΙΓΡΑΦΗΣ
    public void setDescription(String x){
        description = x;
    }

    //ΕΠΙΣΤΡΟΦΗ ΣΥΝΤΕΤΑΓΜΕΝΩΝ
    public double getLan(){
        return lan;
    }
}
```

```

        public double getLon(){
            return lon;
        }
        //ΕΠΙΣΤΡΟΦΗ ΠΕΡΙΓΡΑΦΗΣ
        public String getDescription(){
            return description;
        }
        //ΜΕΤΑΤΡΟΠΗ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΕ STRING
        public String toStringFile(){
            return Double.toString(lon) + ":" + Double.toString(lon) + ":"
+ description + "\n";
        }
    }
}

```

## Αναφορές

[1] Brad Reed, A Brief History of Smartphones, [http://www.techhive.com/article/199243/a\\_brief\\_history\\_of\\_smartphones.html](http://www.techhive.com/article/199243/a_brief_history_of_smartphones.html)

[2] Pei Zheng, Lionel Ni, Smart Phone and Next Generation Mobile Computing, <http://books.google.gr/books?id=iyEP-CX5V0oC&pg=PA24&lpg=PA24&dq=smart+phpnos+history&source=bl&ots=NwMbX8Kas4&sig=L8l39iV5Or7lpcXyxux51eWwoW8&hl=el&sa=X&ei=rZl0U57LHqqQ0AWei4DQDg&ved=0CJkBEOgBMAg#v=onepage&q=smart%20phpnos%20history&f=false>

[3] Saikiran Reddy Sama, A Brief Description On Android OS, <http://technouniversityworld.blogspot.gr/2012/03/brief-description-on-android-os.html>

[4] Android, Developing Accessible Applications, <http://developer.android.com/training/accessibility/accessible-app.html>

[5] Kent German, A brief history of Android phones, visual.ly, The History of Android, <http://www.cnet.com/news/a-brief-history-of-android-phones/>

[6] Visual.ly, Android History, <http://visual.ly/history-android>

[7] Apple, iOS Technology Overview, <https://developer.apple.com/library/ios/documentation/miscellaneous/conceptual/iphoneostechoverview/iOSTechOverview.pdf>

[8] Ben Mirris, The Symbian OS Architecture Sourcebook: Design and Evolution of a Mobile Phone OS (Symbian Press), <http://www.amazon.com/The-Symbian-Architecture-Sourcebook-Evolution/dp/0470018461>

[9] O'Reilly Media, Inc., Developer Interior, <http://chimera.labs.oreilly.com/books/1234000001853/index.html>

[10] Eclipse, Eclipse Articles, Tutorials, Demos, Books, and More, [http://wiki.eclipse.org/Eclipse\\_Articles,\\_Tutorials,\\_Demos,\\_Books,\\_and\\_More](http://wiki.eclipse.org/Eclipse_Articles,_Tutorials,_Demos,_Books,_and_More)

[11] Vogella.com, Android Development - Tutorial, <http://www.vogella.com/tutorials/Android/article.html>

[12] Jimmy McGee, How to Build an Android App Part 1: Setting up Eclipse and Android SDK – XDA TV

<http://www.xda-developers.com/xda-tv-2/how-to-build-an-android-app-part-1-setting-up-eclipse-and-android-sdk-xda-tv/>

[13] androidhive.info, Android working with Google Maps V2, <http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/>

[14] <http://www.android.com>, Android 4.3, Jelly Bean, <http://www.android.com/about/jelly-bean/>

[15] <http://www.android.com>, Android 4.4, Kit Kat, <http://www.android.com/versions/kit-kat-4-4/>

[16] <http://AndroLib.com>, "Android App Stats".

[17] <http://techcrunch.com>, Leena Rao, "Google: 3 Billion Android Apps Installed; Downloads Up 50 Percent From Last Quarter".

[18] <http://www.ibm.com>, Android NDK, <http://www.ibm.com/developerworks/opensource/tutorials/os-androidndk/os-androidndk-pdf.pdf>