



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**ΣΧΟΛΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ, ΓΕΩΓΡΑΦΙΑΣ ΚΑΙ
ΕΦΑΡΜΟΣΜΕΝΩΝ ΟΙΚΟΝΟΜΙΚΩΝ**

ΤΜΗΜΑ ΓΕΩΓΡΑΦΙΑΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ‘ ΕΦΑΡΜΟΣΜΕΝΗ
ΓΕΩΓΡΑΦΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΧΩΡΟΥ ‘**

ΚΑΤΕΥΘΥΝΣΗ ‘ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ ‘

**‘Ανάπτυξη εφαρμογής σε περιβάλλον Android για την παρουσίαση του
Χαροκόπειου Πανεπιστημίου’**

Διπλωματική εργασία του Καλλιαντά Δημήτριου

A.M 216306

Αθήνα, Ιούνιος 2019



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**ΣΧΟΛΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ, ΓΕΩΓΡΑΦΙΑΣ ΚΑΙ
ΕΦΑΡΜΟΣΜΕΝΩΝ ΟΙΚΟΝΟΜΙΚΩΝ**

ΤΜΗΜΑ ΓΕΩΓΡΑΦΙΑΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ‘ ΕΦΑΡΜΟΣΜΕΝΗ
ΓΕΩΓΡΑΦΙΑ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΧΩΡΟΥ ‘**

ΚΑΤΕΥΘΥΝΣΗ ‘ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ ‘

Τριμελής Εξεταστική Επιτροπή

Καλογήρου Σταμάτης (Επιβλέπων)

Επίκουρος Καθηγητής, Τμήμα Γεωγραφίας, Χαροκόπειο Πανεπιστήμιο

Παρχαρίδης Ισαάκ

Καθηγητής, Τμήμα Γεωγραφίας, Χαροκόπειο Πανεπιστήμιο

Μαλινδρέτος Γεώργιος

Επίκουρος Καθηγητής, Τμήμα Γεωγραφίας, Χαροκόπειο Πανεπιστήμιο

Ο Καλλιαντάς Δημήτριος του Παναγιώτη

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Η παρούσα πτυχιακή εργασία αφιερώνεται στην Όλγα & τη Ζωή

Τεχνολογία είναι η δυνατότητα να διευθετούμε τον κόσμο κατά τέτοιο τρόπο, ώστε να μη γευόμαστε την εμπειρία του.

Rollo May, 1909-1994

ΕΥΧΑΡΙΣΤΙΕΣ

Ξεκινώντας, θα ήθελα να ευχαριστήσω τους διδάσκοντες στο Π.Μ.Σ για το υψηλό επίπεδο διδασκαλίας που παρείχαν, μεταδίδοντας στους συμμετέχοντες νέες ιδέες και γνώσεις σε ένα πολύ ξεχωριστό πεδίο σπουδών.

Ιδιαίτερη αναφορά θα ήθελα να κάνω στους συμφοιτητές μου στο Π.Μ.Σ ,γιατί παρότι ξεκινήσαμε σαν άγνωστοι την πορεία μας φτάσαμε να έχουμε ιδιαίτερους φιλικούς δεσμούς.

Η υποστήριξη των συνεργατών μου στη Geosynergy, για άλλη μια φορά, με βοήθησε να ανταπεξέλθω τόσο στις ακαδημαϊκές όσο και στις επαγγελματικές υποχρεώσεις, θέλοντας ειδικότερα να μνημονεύσω τη βοήθεια που είχα από την κ. Ειρήνη Μοσχούλα.

Σε όλες τις πτυχές της ζωής μου έχω την αμέριστη συμπαράσταση της οικογένειάς μου και τους ευχαριστώ μέσα από την καρδιά μου για αυτό.

Θα ήθελα, τέλος, να ευχαριστήσω τον κ. Καλογήρου για τη συνεργασία που είχαμε όλα τα χρόνια, ήδη από το προπτυχιακό επίπεδο, καθώς και για τις πολύτιμες συμβουλές του που με βοήθησαν σε όλη τη φοιτητική μου πορεία.

Περιεχόμενα

Περιεχόμενα.....	6
Περίληψη	7
Περίληψη στα Αγγλικά- Abstract	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	9
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	11
ΕΙΣΑΓΩΓΗ	13
ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ.....	14
1. Έξυπνα κινητά τηλέφωνα και ακαδημαϊκές εφαρμογές	14
2. Ορισμός Λειτουργικών Συστημάτων Κινητών Συσκευών	15
3. Εξέλιξη και ιστορική αναδρομή του Android	21
4. Η Αρχιτεκτονική του Android	36
5. Δομικά χαρακτηριστικά εφαρμογών	39
7. Ασφάλεια στο Android	41
ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ	42
1. Εργαλεία ανάπτυξης λογισμικού για Android	42
2. Γενικά το Android Studio.....	44
3. Android software development kit (SDK).....	44
4. Χαρακτηριστικά του Android Studio IDE.....	45
5. Εργαλεία ανάπτυξης λογισμικού Android (Android Software Development Kit SDK)	47
6. Ανάπτυξη εφαρμογής MyHUAApp.....	51
ΣΥΜΠΕΡΑΣΜΑΤΑ	64
ΒΙΒΛΙΟΓΡΑΦΙΑ	65
ΠΑΡΑΡΤΗΜΑ	67

Περίληψη

Η παρούσα διπλωματική εργασία έγινε για την ανάπτυξη εφαρμογής έξυπνων συσκευών λειτουργικού Android με τη χρήση του λογισμικού Android Studio και των υποστηρικτικών εργαλείων, Android API (Application Programming Interface) και Android SDK (System Development Kit).

Αρχικά, γίνεται παρουσίαση και ιστορική αναδρομή στα λειτουργικά συστήματα φορητών συσκευών και ιδιαίτερα στα χαρακτηριστικά και τις εκδόσεις του λειτουργικού Android.

Επόμενο βήμα, είναι η ανάλυση της αρχιτεκτονικής δομής του λειτουργικού και των απαιτήσεων που υπάρχουν για την ανάπτυξη εφαρμογών και στο τέλος γίνεται περιγραφή της ανάπτυξης μιας εφαρμογής με τη χρήση αυτών των εργαλείων.

Με την αναπτυχθείσα εφαρμογή ο χρήστης θα μπορεί να έχει άμεση και γρήγορη πρόσβαση σε χρήσιμες πληροφορίες σχετικά με το Χαροκόπειο Πανεπιστήμιο (πληροφορίες για τις σχολές, τα μαθήματα, τους καθηγητές, τα στοιχεία επικοινωνίας και πρόσβασης του Ιδρύματος κλπ)

Λέξεις κλειδιά: Λειτουργικό σύστημα Android, ανάπτυξη εφαρμογής, λογισμικό Android studio, Χαροκόπειο Πανεπιστήμιο

Περίληψη στα Αγγλικά- Abstract

The scope of this dissertation is to develop an Android application for smart devices, using Android studio software and its support tools, Android API (Application Programming Interface) and Android SDK (System Development Kit).

Firstly there is a presentation and a background perspective on mobile devices operating systems, especially at Android characteristics and editions.

The next step is to analyze the architectural structure of the operating system and the requirements that exist for the development of applications and finally to describe the development of an application using these tools.

With the developed application, the user can have direct and quick access to useful information about Harokopio University (information on the faculties, courses, teachers, contact and access information of the Foundation etc.)

Keywords: Android OS, Android application, Android studio software, Harokopio University

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Ποσοστό χρήσης λειτουργικών συστημάτων για φορητές συσκευές

Εικόνα 2. Logo Android Banana Bread

Εικόνα 3. Logo Android Cupcake

Εικόνα 4 . Logo Android Donut

Εικόνα 5. Logo Android Éclair

Εικόνα 6. Logo Android Froyo

Εικόνα 7. Logo Android Gingerbread

Εικόνα 8. Logo Android

Εικόνα 9. Logo Android Icecream Sandwich

Εικόνα 10. Logo Android Jellybean

Εικόνα 11. Logo Android KitKat

Εικόνα 12. Logo Android Lollipop

Εικόνα 13. Logo Android Marshmallow

Εικόνα 14. Logo Android Nougat

Εικόνα 15. Logo Android Oreo

Εικόνα 16. Logo Android Pie

Εικόνα 17. Logo Android Q

Εικόνα 18. Αρχιτεκτονική Android

Εικόνα 19. Κεντρικό μενού εφαρμογής

Εικόνα 20. Επιλογή 'Γενικών Πληροφοριών'

Εικόνα 21. Μενού 'Τμημάτων'

Εικόνα 22. Πλοήγηση επιλογής 'Γεωγραφίας'

Εικόνα 23. Πλοήγηση επιλογής 'Πληροφορικής'

Εικόνα 24. Πλοήγηση επιλογής 'Διαιτολογίας'

Εικόνα 25. Πλοήγηση επιλογής 'Οικ. Οικονομίας'

Εικόνα 26. Μενού 'Υπηρεσιών'

Εικόνα 27. Πλοήγηση επιλογής 'Επικοινωνίας'

Εικόνα 28. Μενού 'Πλοήγηση'

Εικόνα 29. Πλοήγηση επιλογής 'Campus 3-D Map'

Εικόνα 30. Πλοήγηση επιλογής 'Take me to HUA'

Εικόνα 31. Πλοήγηση επιλογής '112-SOS'

Εικόνα Π1: Στάδια εγκατάστασης android

Εικόνα Π2: Βήματα παραμετροποίησης διεπαφής

Εικόνα Π3: Διαδικασία δημιουργίας νέου σχεδίου εφαρμογής

Εικόνα Π4: Κεντρική εικόνα πλατφόρμας ανάπτυξης κώδικα

Εικόνα Π5: Επιλογή πρόσθετων πακέτων SDK

Εικόνα Π6: Ρύθμιση Android Virtual Device Manager

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

A.E.I	Ανώτατα Εκπαιδευτικά Ιδρύματα
A.Π.Θ	Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
A.T.E.I	Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δ.Ε.Π	Διδακτικό Ερευνητικό Προσωπικό
Π.Μ.Σ	Πρόγραμμα Μεταπτυχιακών Σπουδών
A.P.I	Application Programming Interface
A.P.K	Android PacKage
C.D.M.A	Code-Division Multiple Access
C.S.S	Cascading Style Sheet
C.2.D.M	Cloud to Device Messaging
D.N.S	Domain Name System
G.I.S	Geographic Information Systems
GLO.NA.S.S	Global Navigation Satellite System
G.P.S	Global Positioning System
H.E.I.F	High Efficiency Image File Format
H.T.M.L	Hyper-Text Markup Language
I.D.E	Integrated Development Environment
J.I.T	Just In Time
M.M.S	Multimedia Messaging Service
N.D.K	Native Development Kit
N.F.C	Near-Field Communication
O.H.A	Open Handset Alliance
O.S	Operating System

P.D.A	Personal Digital Assistant
Q.M.L	Qt Modeling Language
S.D CARD	Secure Digital Card
S.D.K	Software Development Kit
S.M.S	Short Message Service
S.Q.L	Structured Query Language
S.S.D	Solid State Disk
T.L.S	Transport Layer Security
U.S.B	Universal Serial Bus
VoIP	Voice over Internet Protocol
Wi-Fi	Wireless Fidelity

ΕΙΣΑΓΩΓΗ

Σκοπός

Σκοπός της μεταπτυχιακής διπλωματικής εργασίας είναι να αναπτυχθεί μια εφαρμογή σε περιβάλλον Android, που θα παρουσιάζει το Χαροκόπειο Πανεπιστήμιο και θα παρέχει χρήσιμες πληροφορίες στους χρήστες της.

Στόχοι αυτής της προσπάθειας αποτελούν η αξιοποίηση των δυνατοτήτων που παρέχουν οι έξυπνες συσκευές (smartphones, tablets κλπ) σε θέματα πληροφόρησης και ενημέρωσης, η εμβάθυνση γνώσεων σε ανάπτυξη εφαρμογών με χρήση κώδικα, η χρήση διαφορετικών γνώσεων (διαχείριση βάσεων δεδομένων, G.I.S, G.P.S, κλπ) που ελήφθησαν κατά την παρακολούθηση του Π.Μ.Σ, καθώς και η μελλοντική έκδοση εφαρμογής για τελικούς χρήστες.

Αρχικά πραγματοποιήθηκε αναζήτηση της σχετικής βιβλιογραφίας, ακολούθησε η συλλογή από έρευνα πεδίου και συνεντεύξεις με άτομα της πανεπιστημιακής κοινότητας (μέλη ΔΕΠ, διοικητικό προσωπικό, υπ. Διδάκτορες, φοιτητές, υποστηρικτικό προσωπικό), του Χαροκόπειου πανεπιστημίου.

Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε στην εργασία περιλαμβάνει τα εξής στάδια:

Α): Μελέτη του θέματος από ελληνική και ξένη βιβλιογραφία (επιστημονικά άρθρα και δημοσιεύσεις, ακαδημαϊκά συγγράμματα, εκδόσεις λογισμικού κλπ).

Β): Επιτόπια έρευνα στο Χαροκόπειο πανεπιστήμιο για συλλογή πραγματικού χρόνου (real- time) στοιχείων σχετικά με τις απαιτήσεις των χρηστών.

Γ): Ανάπτυξη σχεδίου εργασίας - διαχωρισμός σε τρία (3) κύρια μέρη (θεωρητικό, εμπειρικό- πρακτικό και παράθεσης συμπερασμάτων - προτάσεων).

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

1. Έξυπνα κινητά τηλέφωνα και ακαδημαϊκές εφαρμογές

Η χρήση λειτουργικών συστημάτων (Operating Systems) που φέρουν 'έξυπνα χαρακτηριστικά' έχει επεκταθεί στην πλειοψηφία των κινητών συσκευών που χρησιμοποιεί η ακαδημαϊκή κοινότητα. Το γεγονός αυτό έχει συντελέσει στην ανάπτυξη διαφόρων χρηστικών εφαρμογών, διαθέσιμων στην ψηφιακή εφαρμογή Google Play, όπως :

- Uniway (λήψεις 10.000+), εφαρμογή που παρέχει πληροφορίες για τις σπουδές έχοντας άμεση πρόσβαση στα επίσημα συστήματα έξι (6) ΑΕΙ & ΑΤΕΙ- ΟΧΙ Χαροκόπειο
- Ηλεκτρονική γραμματεία (λήψεις 50.000+), εφαρμογή αντίστοιχη με την παραπάνω, με πρόσβαση σε τριανταένα (31) Πανεπιστήμια & ΑΤΕΙ- ΟΧΙ Χαροκόπειο
- Open e-class (λήψεις 10.000+), εφαρμογή για σύνδεση με το e-class
- UthMe (λήψεις 10.000+), εφαρμογή παροχής πολλαπλών πληροφοριών για το Πανεπιστήμιο Θεσσαλίας
- MyAuth (λήψεις 10.000+), εφαρμογή παροχής πολλαπλών πληροφοριών για το Αριστοτέλειο Πανεπιστήμιο
- ΑΠΘ mobile (λήψεις 10.000+), εφαρμογή παροχής πολλαπλών πληροφοριών για το Αριστοτέλειο Πανεπιστήμιο, όχι μόνο για φοιτητές αλλά και διδάσκοντες ή/ και επισκέπτες

Ο αριθμός λήψεων που αναφέρθηκε παραπάνω, δείχνει τη σημαντική ζήτηση που υπάρχει από την πλευρά των τελικών χρηστών για τη μέγιστη δυνατή αξιοποίηση των 'έξυπνων' χαρακτηριστικών που φέρουν οι κινητές συσκευές.

Παράλληλα εντοπίζεται η απουσία υιοθέτησης ή συμμετοχής του Χαροκόπειου πανεπιστημίου σε εφαρμογές νέων τεχνολογιών, πιθανότατα λόγω του αρκετά περιορισμένου αριθμού χρηστών σε σχέση με τα υπόλοιπα ιδρύματα.

Είναι δεδομένο πλέον πως η χρήση των εφαρμογών (applications) προτιμάται για άμεση πρόσβαση σε μια σειρά πληροφοριών και δεδομένων σε σχέση με τους συμβατικούς τρόπους (επίσκεψη, τηλέφωνο, αναζήτηση στο διαδίκτυο κλπ), οπότε η ανάπτυξη και προώθηση τέτοιων εφαρμογών θα απασχολήσει σύντομα το σύνολο των ιδρυμάτων τριτοβάθμιας εκπαίδευσης.

2. Ορισμός Λειτουργικών Συστημάτων Κινητών Συσκευών

Οι σύγχρονοι υπολογιστές αποτελούνται πολλούς επεξεργαστές, μεγάλη κύρια μνήμη, δίσκους SSD, πληκτρολόγιο, ποντίκι, οθόνη, διασυνδέσεις δικτύου και διάφορες άλλες συσκευές εισόδου/εξόδου όπου όλα αυτά υπάγονται σε ένα διαλειτουργικό σύστημα.

Η διαχείριση όλων αυτών των στοιχείων, η χρήση των βέλτιστων δυνατοτήτων και η ορθή κατανομή πόρων είναι ένα εξαιρετικά περίπλοκο και δύσκολο έργο που το διαχειρίζεται ένας τύπος λογισμικού που ονομάζεται Λειτουργικό Σύστημα (Operating System).

Στην ουσία, η ύπαρξη του λειτουργικού συστήματος παρέχει το μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού (software) και υλικού (hardware), διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή.

Η διαχείριση του υλικού αποτελεί ίσως την κρίσιμότερη και σημαντικότερη διεργασία του λειτουργικού συστήματος, μιας και βοηθάει τον χρήστη να αποφύγει τον άμεσο και επίπονο χειρισμό του υπολογιστή ενώ παράλληλα καθιστά φιλικότερο τον προγραμματισμό λογισμικού.

Πέρα από τη φιλικότητα στη χρήση, το κόστος απόκτησης- χρήσης του λειτουργικού έχει μεγάλη σημασία και στην ουσία οι δυο αυτοί λόγοι καθορίζουν και το επίπεδο του ανταγωνισμού μεταξύ των εταιρειών παραγωγής και διανομής λειτουργικών συστημάτων.

Το βασικό τμήμα ενός λειτουργικού συστήματος το οποίο αλληλεπιδρά στενά με το υλικό του υπολογιστή είναι ο πυρήνας (Kernel), καθώς είναι υπεύθυνος για την αυτόματη διαχείριση των πόρων του συστήματος, την κατανομή τους στις εκτελούμενες εφαρμογές, την απρόσκοπτη εκτέλεση των τελευταίων και την επικοινωνία μεταξύ υλικού και λογισμικού.

Ειδικά τα λειτουργικά συστήματα κινητών συσκευών συνδυάζουν χαρακτηριστικά ενός προσωπικού υπολογιστή μαζί με άλλα χαρακτηριστικά και πόρους, χρήσιμα για την κινητή ή φορητή χρήση, όπως η οθόνη αφής (Touch Screen), η τηλεφωνία (Cellular), το Bluetooth, η αναγνώριση βιομετρικών χαρακτηριστικών (Finger print, Face recognition κλπ), η Ασύρματη Δικτύωση (Wi-Fi), ο δορυφορικός εντοπισμός σήματος/πλοηγός (GPS/GLONASS/GALILEO/BAIDU navigation), η φωτογραφική μηχανή (Camera), η βίντεο κάμερα (Video Camera), η αναγνώριση φωνής (Speech Recognition), η ηχογράφηση φωνής (Voice Recorder), η αναπαραγωγή μουσικής (Music Player), οι ασύρματες επικοινωνίες μικρής εμβέλειας (Near Field Communications NFC), η δυνατότητα προβολής σε άλλες συσκευές (Casting) κλπ.

Σύμφωνα με την εταιρεία ανάπτυξης (Google) το Android είναι ένα λειτουργικό σύστημα, ανοικτού κώδικα και αποτελεί μια ολοκληρωμένη, ανοικτή και ελεύθερα διαθέσιμη πλατφόρμα, η οποία περιλαμβάνει, πέρα από τον πυρήνα του λειτουργικού συστήματος (kernel), ένα σύνολο βιβλιοθηκών και βασικών εφαρμογών.

Αρχικά, το λειτουργικό σύστημα Android αναπτύχθηκε το 2003 από την Android Inc., μία start-up εταιρεία που στη συνέχεια εξαγοράστηκε από την Google, η οποία με τη σειρά της ίδρυσε το 2007 μία σύμπραξη με το όνομα Open Handset Alliance.

Σύμφωνα με το ιστορικό της, η νέα αυτή εταιρεία αποτελείται από 84 μέλη και μελετά την ανάπτυξη ανοικτών προτύπων για κινητές συσκευές, έχοντας υπό την ευθύνη της και την συντήρηση και εξέλιξη της πλατφόρμας Android, ενώ έχουν δημοσιεύσει το μεγαλύτερο μέρος του κώδικα Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού (DiMarzio, 2016).

Η βάση της λειτουργίας του βρίσκεται σε έναν τροποποιημένο πυρήνα του Linux και προσφέρει μία ανοικτή πλατφόρμα για απλούς ή προχωρημένους χρήστες που θέλουν να

κάνουν χρήση φορητών συσκευών εξοπλισμένων με δυνατότητες όπως GPS, κάμερα, οθόνη αφής, πολυμεσικές λειτουργίες, αισθητήρες διαφόρων τύπων κτλ.

Το λειτουργικό Android λόγω του φιλικού περιβάλλοντος και της ανοικτής φύσης του έχει προσελκύσει πολλούς καταναλωτές και developers, καθώς το πλαίσιο ανάπτυξης εφαρμογών (application development framework) του Android επιτρέπει σε χρήστες διαφόρων επιπέδων γνώσης να δημιουργήσουν τις δικές τους εφαρμογές για μεγάλο εύρος συσκευών (smartphone, tablet, smart TV, smartwatch κλπ). Αυτό οδηγεί στο να έχει το λειτουργικό σχεδόν απεριόριστη λειτουργικότητα χρησιμοποιώντας μια ποικιλία από βιβλιοθήκες και εργαλεία αυξάνοντας έτσι την παραγωγικότητά τους καθώς και την ποιότητα των εφαρμογών (Burnette, 2015).

Πριν από την εμφάνιση του Android, υπήρχαν διάφορα άλλα λειτουργικά συστήματα για κινητές συσκευές. Οι πλατφόρμες αυτές ήταν κατά κανόνα κλειστές, απευθύνονταν σε συγκεκριμένες συσκευές και για την ανάπτυξη του κατάλληλου λογισμικού απαιτούνταν η χρήση κλειστών εργαλείων. Συχνά επίσης δεν υποστηριζόταν, επί ίσοις όροις, η ανάπτυξη εφαρμογών από τρίτους κατασκευαστές (Reto, 2010).

Παραθέτοντας μια συνοπτική ιστορική αναδρομή στην εξέλιξη των Κινητών Συσκευών και των Λειτουργικών Συστημάτων τους (Silberschatz et al, 2005 & Tanenbaum et al, 2009) βλέπουμε τις παρακάτω ημερομηνίες σταθμούς:

- 1973–1993: Τα κινητά τηλέφωνα χρησιμοποιούν ενσωματωμένα συστήματα (Embedded Systems) για τον έλεγχο της λειτουργίας τους.
- 1995: Το πρώτο έξυπνο τηλέφωνο (Smartphone), IBM Simon, έχει οθόνη αφής, λειτουργία e-mail καθώς και χαρακτηριστικά PDA (Personal Digital Assistant).
- 1996: Το PDA Palm Pilot 1000 με το Λειτουργικό Σύστημα Palm OS.
- 1996: Παρουσιάζονται οι πρώτοι υπολογιστές χειρός (Handheld PC) με το λειτουργικό σύστημα Windows CE.

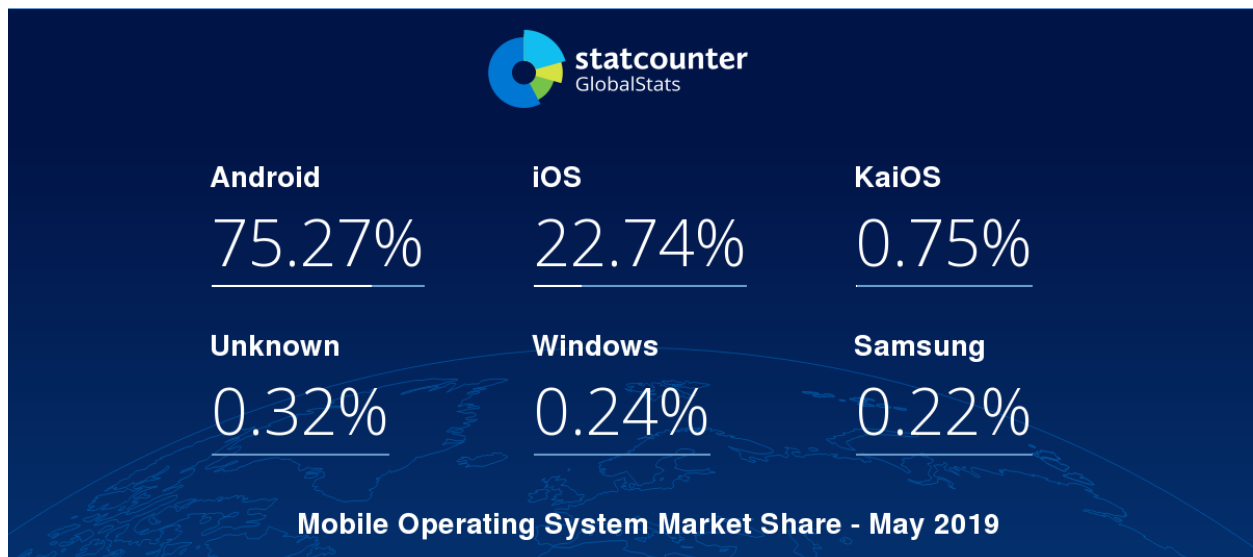
- 1999: Παρουσιάζεται επίσημα το λειτουργικό σύστημα Nokia S40 μαζί με τη συσκευή κινητής τηλεφωνίας Nokia 7110.
- 2000: Το Symbian γίνεται το πρώτο μοντέρνο λειτουργικό σύστημα κινητών τηλεφώνων εγκατεστημένο πάνω στο έξυπνο τηλέφωνο Ericsson R380.
- 2001: Το Kyocera 6035 είναι το πρώτο έξυπνο τηλέφωνο με το λειτουργικό σύστημα Palm OS.
- 2002: Παρουσιάζονται οι πρώτοι υπολογιστές τσέπης (Pocket PC) της Microsoft με λειτουργικό σύστημα Windows CE
- 2002: Η BlackBerry κυκλοφορεί το πρώτο της έξυπνο τηλέφωνο.
- 2005: Η Nokia παρουσιάζει το λειτουργικό σύστημα Maemo OS πάνω στο διαδικτυακό (Internet) tablet N770.
- 2007: Η Apple παρουσιάζει το κινητό τηλέφωνο iPhone με το λειτουργικό σύστημα iOS.
- 2007: Γίνεται ο σχηματισμός της Open Handset Alliance (OHA) από την Google, την HTC, την Sony, την Dell, την Intel, την Motorola, την Samsung, την LG κ.ο.κ.
- 2008: Ο OHA απελευθερώνει το Android 1.0 (με βάση το Linux Kernel) με το HTC Dream (T-Mobile G1) ως το πρώτο τηλέφωνο Android.
- 2009: Η Palm παρουσιάζει με το λειτουργικό σύστημα webOS το Palm Pre. Το 2012 οι webOS βασισμένες συσκευές σταμάτησαν να πωλούνται.
- 2009: Η Samsung δημοσιεύει το λειτουργικό σύστημα Bada OS με την παρουσίαση της συσκευής Samsung S8500.
- 2010: Τα τηλέφωνα με λειτουργικό σύστημα Windows Phone OS παρουσιάζονται, αλλά δεν είναι συμβατά με το λειτουργικό προηγούμενης έκδοσης Windows Mobile OS.

- 2011: Το λειτουργικό σύστημα MeeGo βασισμένο σε Linux που συνδυάζει το Maemo και το Moblin, παρουσιάζεται μαζί με το Nokia N9, σε συνεργασία της Nokia, της Intel και του Linux Foundation.
- 2011: Η Samsung, η Intel και το Linux Foundation ανακοινώνουν τον Σεπτέμβρη, ότι οι προσπάθειές τους θα μετατοπιστούν από Bada και MeeGo σε Tizen κατά τη διάρκεια του 2011-2012.
- 2011: Τον Οκτώβρη του 2011 ανακοινώνεται το Mer, ένα κεντριοποιημένο project γύρω από ένα μέγα-φορητό πυρήνα Linux +HTML5/QML/JavaScript για την κατασκευή προϊόντων που προέρχονται από τη βάση κώδικα του λειτουργικού συστήματος MeeGo.
- 2012: Η Mozilla ανακοινώνει τον Ιούλιο του 2012 ότι το project γνωστό ως Boot to Gecko (το οποίο χτίστηκε πάνω στον πυρήνα Android Linux χρησιμοποιώντας οδηγούς του Android, ωστόσο δεν χρησιμοποιεί τη γλώσσα Java του Android) είναι πλέον το λειτουργικό σύστημα Firefox OS.
- 2013: Η Canonical ανακοινώνει το Ubuntu Touch, μια έκδοση των Linux σχεδιασμένη συγκεκριμένα για έξυπνα τηλέφωνα. Χτίστηκε πάνω στον πυρήνα Android Linux χρησιμοποιώντας οδηγούς του Android, αλλά δεν χρησιμοποιεί τη γλώσσα Java του Android.
- 2013: Η BlackBerry κυκλοφορεί το καινούργιο της λειτουργικό σύστημα BlackBerry 10.
- 2013: Η Google κυκλοφορεί το λειτουργικό σύστημα Android έκδοση KitKat 4.4.
- 2014: Η Microsoft το Φεβρουάριο του 2014 κυκλοφορεί το λειτουργικό σύστημα Windows Phone 8.1.
- 2014: Η Apple το Σεπτέμβρη του 2014 κυκλοφορεί το λειτουργικό iOS 8.
- 2014: Η BlackBerry το Σεπτέμβρη του 2014 κυκλοφορεί το λειτουργικό σύστημα BlackBerry 10.3 ενοποιημένο με το Amazon Appstore.

- 2014: Η Google το Νοέμβρη του 2014 κυκλοφορεί το λειτουργικό σύστημα Android 5.0 έκδοση "Lollipop".
- 2015: Η Apple το Σεπτέμβρη του 2015 κυκλοφορεί το λειτουργικό σύστημα iOS 9.
- 2015: Η Google το Σεπτέμβρη του 2015 κυκλοφορεί το λειτουργικό σύστημα Android 6.0 έκδοση "Marshmallow".
- 2016: Η Google κυκλοφορεί το λειτουργικό σύστημα Android 7.0 έκδοση "Nougat".
- 2017: Η Apple το 2017 κυκλοφορεί το λειτουργικό σύστημα iOS 10.
- 2017: Η Google Android 8.0 έκδοση "Oreo".
- 2018: Η Google Android 9.0 έκδοση "Pie".
- 2019: Η Apple το 2019 κυκλοφορεί το λειτουργικό σύστημα iOS 12.

Στην εικόνα 1 παρουσιάζονται τα ποσοστά χρήσης των λειτουργικών συστημάτων για φορητές συσκευές και φαίνεται πως το μεγαλύτερο ποσοστό των συσκευών, της τάξης του 75,27 %, χρησιμοποιούν Android, ενώ ακολουθούν οι συσκευές που χρησιμοποιούν το iOS της Apple με ποσοστό 22,74%, ενώ τα υπόλοιπα λειτουργικά συστήματα καταλαμβάνουν ένα πολύ μικρό μερίδιο της αγοράς.

Λαμβάνοντας υπόψιν το πολύ μεγάλο ποσοστό χρήσης του Android, καθώς και το γεγονός ότι το λειτουργικό σύστημα iOS είναι ένα απόλυτα κλειστό λογισμικό μιας και απευθύνεται σε συσκευές ενός συγκεκριμένου κατασκευαστή (Apple), είναι εύκολο να κατανοηθούν τα πλεονεκτήματα της χρήσης του λειτουργικού συστήματος Android, καθώς και της ανάπτυξη εφαρμογών για συσκευές σε αυτό το λειτουργικό σύστημα.



Εικόνα 1. Ποσοστό χρήσης λειτουργικών συστημάτων για φορητές συσκευές (πηγή: *statcounter.com*, Μάιος 2019)

3. Εξέλιξη και ιστορική αναδρομή του Android

Ο χαρακτηρισμός του Android ως ανοιχτό λογισμικό έχει συμβάλει στη διαρκή ανάπτυξη και βελτίωση του λειτουργικού καθώς υπάρχουν ήδη 17 κύριες εκδόσεις μέσα σε 11 χρόνια.

Η κωδική ονομασία των διαφόρων εκδόσεων του Android προέρχεται από ονομασίες γλυκών (Android.com). Οι εκδόσεις αυτές είναι:

Android 1.1 Banana Bread



Εικόνα 2. Android Banana Bread (πηγή: Android.com)

Η αρχική έκδοση 1.1 κυκλοφόρησε το Φεβρουάριο του 2009, αποτελώντας ένα σημαντικό γεγονός στην ιστορία των λειτουργικών συστημάτων φορητών συσκευών.

Η επίσημηση (1.1) έδειχνε στην αγορά ότι το Android θα επιδεχόταν αναβαθμίσεις (κάτι που ανταγωνιστικές πλατφόρμες δεν υποστήριζαν ακόμη) για τη βελτίωση της εμπειρίας του χρήστη (<https://developer.android.com>)

Android 1.5 Cupcake



Εικόνα 3. Logo Android Cupcake (πηγή: Android.com)

Η έκδοση κυκλοφόρησε στις 30 Απριλίου 2009, με επιπλέον χαρακτηριστικά ένα νέο εικονικό πληκτρολόγιο και τη μεταφόρτωση αρχείων video- εικόνων στο Youtube και το Picasa.

Android 1.6 Donut



Εικόνα 4 . Logo Android Donut (πηγή: Android.com)

Η έκδοση παρουσιάστηκε στις 15 Σεπτεμβρίου 2009, με χαρακτηριστικά την επιλογή πολλαπλών αρχείων, τη βελτιωμένη γκαλερί, τη λειτουργία φωτογραφικής μηχανής και τη σύνδεση με τον ιστότοπο εφαρμογών Android Market. Ο χρήστης έχει επίσης τη δυνατότητα για βελτιωμένη αναζήτηση που συμπεριλαμβάνει την αναζήτηση σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο μέσω αρχικής οθόνης.

Επιπρόσθετα χαρακτηριστικά που εισάγονται με αυτήν την ενημερωμένη έκδοση είναι η υποστήριξη για CDMA smartphones, τα επιπλέον μεγέθη οθόνης, ο δείκτης χρήσης της μπαταρίας, και μια μηχανή μετατροπής κειμένου σε ομιλία.

Μετά τη δημόσια κυκλοφορία του Donut οι εταιρείες κινητής τηλεφωνίας έκαναν διαθέσιμη τη μορφή ενημέρωσης over-the-air (OTA) για τα συμβατά έξυπνα τηλέφωνα.



Εικόνα 5. Logo Android Éclair(πηγή: Android.com)

Παρουσιάστηκε στις 26 Οκτωβρίου 2009 και επανεκδόθηκε σε Android 2.1 Eclair τον Ιανουάριο του 2010. Σε αυτή την έκδοση η προεπιλεγμένη αρχική οθόνη του Eclair εμφανίζει το Google Search στην κορυφή της οθόνης και παρουσιάζεται για πρώτη φορά η εισαγωγή ομιλίας σε κείμενο (φωνητική πληκτρολόγηση).

Η εφαρμογή της κάμερας επίσης έχει επανασχεδιαστεί με πολλές νέες λειτουργίες στην κάμερα, συμπεριλαμβανομένων της υποστήριξης φλας, του ψηφιακού zoom, τη λειτουργία σκηνής, την ισορροπία λευκού, το εφέ χρώματος και τη μακρο εστίαση.

Η εφαρμογή 'άλμπουμ φωτογραφιών' περιέχει επίσης βασικά εργαλεία επεξεργασίας φωτογραφιών, ενώ εκτός από τις ζωντανές ταπετσαρίες, επιτρέπει τις κινούμενες εικόνες φόντου στην αρχική οθόνη για να δείξει την κίνηση.

Το Android Eclair προσθέτει επίσης υποστήριξη για επικοινωνία κοντινού πεδίου (NFC), την ικανότητα αναζήτησης όλων των αποθηκευμένων μηνυμάτων SMS και MMS, βελτίωση του Google Maps 3.1.2, και υποστήριξη Exchange για την εφαρμογή Email.

Το λειτουργικό σύστημα παρέχει επίσης βελτιωμένη ταχύτητα πληκτρολόγησης στο εικονικό πληκτρολόγιο, μαζί με την νέα προσβασιμότητα, το ημερολόγιο, και το εικονικό ιδιωτικό δίκτυο APIs.

Για την περιήγηση στο διαδίκτυο, το Android Eclair προσθέτει επίσης την υποστήριξη HTML5, με ανανεωμένο UI πρόγραμμα περιήγησης με μικρογραφίες σελιδοδείκτη και διπλό άγγιγμα για ζουμ.

Android 2.2 Froyo



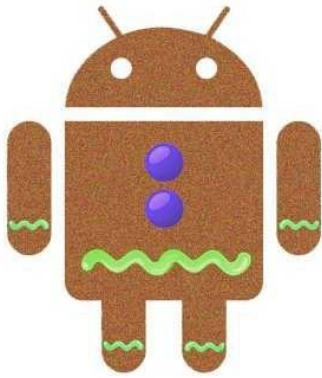
Εικόνα 6. Logo Android Froyo (πηγή: Android.com)

Η παρουσίαση της έκδοσης αυτής έγινε 20 Μαΐου του 2010, κατά τη διάρκεια του συνεδρίου Google I/O 2010. Σε αυτή την έκδοση υπάρχει η υποστήριξη Tethering, η λειτουργία Hotspot, το Adobe Flash 10.1 και τέλος, η δυνατότητα της εγκατάστασης εφαρμογών στην εξωτερική κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη τηλεφώνου.

Επιπλέον, μία από τις πιο εξέχουσες αλλαγές στην κυκλοφορία του Froyo ήταν η σύνδεση με USB και η λειτουργία Wi-Fi hotspot.

Άλλες αλλαγές αφορούσαν στην υποστήριξη για το Android Cloud στα Μηνύματα (C2DM) υπηρεσίας συσκευής, επιτρέποντας τα Push Notifications, ενώ έγιναν πρόσθετες βελτιώσεις της ταχύτητας εφαρμογής, που υλοποιούνται μέσω καθορισμού JIT και εμφανίζεται μέσα σε εφαρμογές όπως top-of-the-screen banners.

Android 2.3 Gingerbread



Εικόνα 7. Logo Android Gingerbread (πηγή: Android.com)

Αυτή η έκδοση παρουσιάστηκε τον Δεκέμβριο του 2010 και επανεκδόθηκε σε Android 2.3.3 τον Φεβρουάριο του 2011. Σε αυτή την έκδοση έχουν γίνει βελτιώσεις στο User Interface κάνοντάς το πιο απλό και ταχύτερο, ενώ υποστηρίζεται πλέον η δυνατότητα για copy-paste σε όλο το σύστημα και η ύπαρξη πολλαπλών καμερών.

Παράλληλα, η κυκλοφορία του Gingerbread εισήγαγε και υποστήριξε την επικοινωνία κοντινού πεδίου (NFC), που χρησιμοποιείται σε πληρωμές μέσω κινητών, και το Session Initiation Protocol(SIP), που χρησιμοποιείται στην τηλεφωνία VoIP τηλεφωνία μέσω internet.

Η διεπαφή χρήστη του Gingerbread είχε τελειοποιηθεί με πολλούς τρόπους, πράγμα που καθιστούσε ευκολότερη και πιο γρήγορη τη χρήση, καθώς και πιο αποδοτική ενεργειακά αφού οι βελτιώσεις στα μενού και τις ρυθμίσεις οδήγησαν στην ευκολότερη πλοήγηση και τον έλεγχο του συστήματος.

Το έξυπνο τηλέφωνο Nexus S, που κυκλοφόρησε το 2010, ήταν το πρώτο τηλέφωνο από τη σειρά Google Nexus που έτρεξε το Gingerbread, και επίσης το πρώτο από τη σειρά με ενσωματωμένη λειτουργία NFC.

Android 3.0 Honeycomb



Εικόνα 8. Logo Android Honeycomb(πηγή: Android.com)

Παρουσιάστηκε το Μάιο του 2011, κάνοντας ντεμπούτο με τη συσκευή Motorola Xoom και προοριζόταν αποκλειστικά για συσκευές tablet, με δυνατότητα υποστήριξης διπύρηνων και τετραπύρηνων επεξεργαστών.

Η έκδοση εισήγαγε το κουμπί recent apps που προσφέρει στο χρήστη την εύκολη προσπέλαση από τη μια εφαρμογή στην άλλη και τη δυνατότητα υποστήριξης 3D γραφικών, Private Browsing και δυνατότητα προβολής εικόνων σε πλήρη οθόνη.

Εκτός από την προσθήκη των ανωτέρω χαρακτηριστικών, το Honeycomb εισήγαγε ένα νέο λεγόμενο «ολογραφικό» θέμα διεπαφής χρήστη και ένα μοντέλο αλληλεπίδρασης που χτίστηκε πάνω στα κύρια χαρακτηριστικά του Android, όπως το multitasking, τις κοινοποιήσεις και τα widgets.

Android 4.0 Ice Cream Sandwich



Εικόνα 9. Logo Android Icecream Sandwich(πηγή: Android.com)

Εμφανίστηκε 19 Οκτωβρίου του 2011 και στηρίζεται στις σημαντικές αλλαγές που έγιναν από τα tablet με την κυκλοφορία του Android Honeycomb, σε μια προσπάθεια να δημιουργήσει μια ενοποιημένη πλατφόρμα για τα έξυπνα τηλέφωνα και τα tablets, και απλουστεύοντας ταυτόχρονα και εκσυγχρονίζοντας τη συνολική εμπειρία του Android γύρω από ένα νέο σύνολο των κατευθυντήριων γραμμών της ανθρώπινης διεπαφής.

Στη συγκεκριμένη διανομή έγινε επανασχεδιασμός με πιο αποδοτικό User Interface που είχε νέα εικονικά κουμπιά αντικαθιστώντας τα αφής ή φυσικά κουμπιά, προσαρμογή των εφαρμογών Gmail- Google Chrome και νέα προσαρμόσιμη οθόνη κλειδώματος, ενώ δινόταν η δυνατότητα μέτρησης δεδομένων ίντερνετ και η δυνατότητα αναγνώρισης προσώπου.

Το Android 4.0 εισήγαγε μια νέα οπτική εμφάνιση με την κωδική ονομασία "Holo", το οποίο είναι χτισμένο γύρω από ένα καθαρότερο, μινιμαλιστικό σχεδιασμό, και μια νέα προεπιλεγμένη γραμματοσειρά που ονομάζεται Roboto.

Παράλληλα, ενσωματώθηκε μια σειρά νέων χαρακτηριστικών, συμπεριλαμβανομένης μιας ανανεωμένης αρχικής οθόνης, η επικοινωνία κοντινού πεδίου (NFC) με έναν άλλο χρήστη, έναν ενημερωμένο web browser, μια νέα εφαρμογή για τις επαφές με την ενσωμάτωση των κοινωνικών δικτύων, τη δυνατότητα πρόσβασης της αναπαραγωγής της κάμερας και τον έλεγχο της μουσικής από την οθόνη κλειδώματος, την οπτική υποστήριξη τηλεφωνητή.

Τέλος, δόθηκε η δυνατότητα της αναγνώρισης προσώπου για ξεκλείδωμα (Face Unlock), η ευχαίρεια των χρηστών να παρακολουθούν και περιορίσουν τη χρήση δεδομένων κινητής και άλλες εσωτερικές βελτιώσεις.

Android 4.1/4.2/4.3 Jellybean



Εικόνα 10. Logo Android Jellybean (πηγή: Android.com)

Παρουσιάστηκε τον Ιούνιο του 2012 και γίνεται η αναβάθμιση του Android Market σε Google Play Store. Η μεγαλύτερη καινοτομία της έκδοσης ήταν ο ψηφιακός βοηθός Google Now ο οποίος μπορούσε να παρέχει πληροφορίες και απαντήσεις για καθημερινά ενδιαφέροντα.

Οι αλλαγές εστίαζαν σε βελτιώσεις στην απόδοση και σχεδιάστηκε έτσι ώστε να δώσει το λειτουργικό σύστημα μια ομαλότερη και πιο δεκτική αίσθηση, βελτιώσεις στο σύστημα κοινοποίησης που επιτρέπει την "επέκταση" των κοινοποιήσεων με κουμπιά δράσης, και άλλες εσωτερικές αλλαγές.

Δύο ακόμη κυκλοφορίες έγιναν υπό την ονομασία Jelly Bean τον Οκτώβριο του 2012 και τον Ιούλιο του 2013 αντίστοιχα, συμπεριλαμβανομένων της 4.2, η οποία περιελάμβανε περαιτέρω βελτιστοποιήσεις, υποστήριξη πολλαπλών χρηστών για tablet, widgets στην οθόνη κλειδώματος, γρήγορες ρυθμίσεις και προφυλάξεις οθόνης, και της 4.3 που περιείχε περαιτέρω βελτιώσεις και ενημερώσεις για την υποκείμενη πλατφόρμα Android.

Android 4.4 KitKat



Εικόνα 11. Logo Android KitKat (πηγή: Android.com)

Παρουσιάστηκε 3 Σεπτεμβρίου του 2013. Σε αυτή την έκδοση υπήρξε μεγάλη βελτίωση του User Interface σε τέτοιο βαθμό ώστε να 'τρέχει' απροβλημάτιστα και σε συσκευές χαμηλών δυνατοτήτων, που διαθέτουν περιορισμένους πόρους.

Android 5.0/5.1 Lollipop

Εικόνα 12. Logo Android Lollipop (πηγή: Android.com)



Παρουσιάστηκε στις 25 Ιουνίου 2014 στο συνέδριο του Google I/O και έγινε διαθέσιμο μέσω της επίσημης over-the-air (OTA) ενημέρωσης στις 12 Νοεμβρίου 2014, για επιλεγμένες συσκευές που 'τρέχουν' διανομές του Android και εξυπηρετούνται από την Google (όπως το Nexus και συσκευές έκδοσης Google Play).

Ο πηγαίος κώδικάς του διατέθηκε στις 3 Νοεμβρίου 2014. Σε αυτή την έκδοση υπάρχει το νέο επίπεδο λειτουργικό σε εμφάνιση, με περισσότερα χρώματα και πολλά εφέ κίνησης.

Μία από τις πιο βασικές αλλαγές στην κυκλοφορία του Lollipop είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μια σχεδιαστική γλώσσα γνωστή ως Material Design, η οποία έγινε για να διατηρήσει μια αίσθηση χαρτιού στο περιβάλλον.

Άλλες αλλαγές περιλαμβάνουν βελτιώσεις στις κοινοποιήσεις, οι οποίες είναι προσβάσιμες από την οθόνη κλειδώματος και εμφανίζονται μέσα σε εφαρμογές. Η Google έκανε επίσης εσωτερικές αλλαγές στην πλατφόρμα, με το Android Runtime (ART) να είναι επίσημο αντικαθιστώντας το Dalvik για βελτιωμένη απόδοση των εφαρμογών, καθώς και με τις αλλαγές που αποσκοπούν στη βελτίωση και τη βελτιστοποίηση της χρήσης της μπαταρίας.

Android 6.0 Marshmallow



Εικόνα 13. Logo Android Marshmallow (πηγή: Android.com)

Παρουσιάστηκε τον Μάιο του 2015 στο συνέδριο του Google I/O, και κυκλοφόρησε επίσημα τον Οκτώβριο του 2015. Στην έκδοση αυτή υπάρχει η δυνατότητα αναγνώρισης δακτυλικού αποτυπώματος για το ξεκλείδωμα της οθόνης ή ακόμα για την πραγματοποίηση πληρωμών.

Το Marshmallow επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη, εισήγαγε μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, νέα APIs για συμφραζόμενα βοηθούς (που χρησιμοποιείται για πρώτη φορά από ένα νέο χαρακτηριστικό - το "Now on Tap"- που παρέχει το πλαίσιο με τα ευαίσθητα αποτελέσματα αναζήτησης), ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του παρασκηνίου όταν μια συσκευή δεν χρησιμοποιείται, εγγενή υποστήριξη για την αναγνώριση δακτυλικών αποτυπωμάτων και USB τύπου-C, τη δυνατότητα να μεταφοράς των δεδομένων και τις εφαρμογές σε μια κάρτα microSD, και άλλες εσωτερικές αλλαγές.

Android 7.0 Nougat



Εικόνα 14. Logo Android Nougat (πηγή: Android.com)

Κυκλοφόρησε για πρώτη φορά ως δοκιμαστική έκδοση (beta build) στις 9 Μαρτίου 2016 και ως επίσημη στις 22 Αυγούστου 2016. Οι συσκευές Nexus ήταν οι πρώτες που αναβαθμίστηκαν με το νέο λογισμικό, αν και το LG V20 ήταν το πρώτο νέο έξυπνο τηλέφωνο που κυκλοφόρησε με Nougat.

Η έκδοση Nougat εισάγει σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξής του, συμπεριλαμβανομένης της δυνατότητας να εμφανίσει πολλαπλές

εφαρμογές στην οθόνη με προβολή διαίρεσης της οθόνης, υποστήριξη για τις απαντήσεις ενσωματωμένη με τις κοινοποιήσεις, καθώς και περιβάλλον Java βασισμένο στο OpenJDK αλλά και υποστήριξη για τις απόδοσεις γραφικών Vulkan API, και απρόσκοπτες ενημερώσεις συστήματος για υποστηριζόμενες συσκευές.

Android 8.0 Oreo



Εικόνα 15. Logo Android Oreo (πηγή: Android.com)

Το Android "Oreo" (κωδική ονομασία Android O) κυκλοφόρησε αρχικά ως έκδοση ανάπτυξης στις 21 Μαρτίου 2017 και τελικά κυκλοφόρησε για τους τελικούς χρήστες στις 21 Αυγούστου, με το Sony Xperia XZ1 να είναι η πρώτη διαθέσιμη συσκευή με προεγκατεστημένο το Oreo.

Η Google παρουσίασε ένα άγαλμα με το μπισκότο Oreo από το οποίο πήρε την ονομασία της η έκδοση, στο πάρκο 14th Street στο Μανχάταν, κοντά στο αρχικό εργοστάσιο Nabisco που δημιούργησε το πρώτο Oreo.

Android 9.0 Pie



Εικόνα 16. Logo Android Pie (πηγή: Android.com)

Το Android "Pie" (γνωστό με την κωδική ονομασία Android P) ανακοινώθηκε για πρώτη φορά από τη Google στις 7 Μαρτίου του 2018, με την τελική έκδοση beta του Android P να κυκλοφορεί στις 25 Ιουλίου 2018.

Το Android "P" κυκλοφόρησε επίσημα στις 6 Αυγούστου 2018 ως "Android 9 Pie" και είναι διαθέσιμη για τις συσκευές Google Pixel και το Essential Phone.

Ενσωματώνει μια σειρά νέων χαρακτηριστικών όπως:

- Νέα διεπαφή χρήστη για το μενού γρήγορων ρυθμίσεων.
- Το ρολόι έχει μετακινηθεί αριστερά από τη γραμμή ειδοποιήσεων.
- Ένα κουμπί "Στιγμιότυπο" έχει προστεθεί στις επιλογές ενέργειας.
- Μια νέα λειτουργία "Κλείδωμα", η οποία απενεργοποιεί τη βιομετρική επαλήθευση μόλις ενεργοποιηθεί, θα απενεργοποιείται μόλις ο χρήστης χρησιμοποιήσει τον κωδικό πρόσβασης για να συνδεθεί.
- Στρογγυλεμένες γωνίες γύρω από την γραφική διεπαφή χρήστη.

- Νέες μεταβάσεις για την εναλλαγή μεταξύ των εφαρμογών ή τις δραστηριότητες εντός των εφαρμογών.
- Πιο πλούσιες ειδοποιήσεις μηνύματος, πλήρους κλίμακας εικόνες και έξυπνες απαντήσεις παρόμοιες με τη νέα εφαρμογή Reply της Google.
- DNS μέσω TLS
- Υποστήριξη HEIF.
- Επανασχεδιασμένος, οριζόντιος πολυεργασιακός εναλλαγέας εφαρμογών με τη μπάρα αναζήτησης Google και το συρτάρι εφαρμογών.
- Προστέθηκε ένα χαρακτηριστικό "Ψηφιακή Ευημερία", το οποίο αποθαρρύνει την υπερβολική χρήση του τηλεφώνου.
- Τοποθετώντας το τηλέφωνο με το πρόσωπο προς τα κάτω θα θέσει τις ειδοποιήσεις σε σίγαση, αλλά επιτρέπει την εμφάνιση ειδοποιήσεων σε καταστάσεις έκτακτης ανάγκης.

Android 10 Q



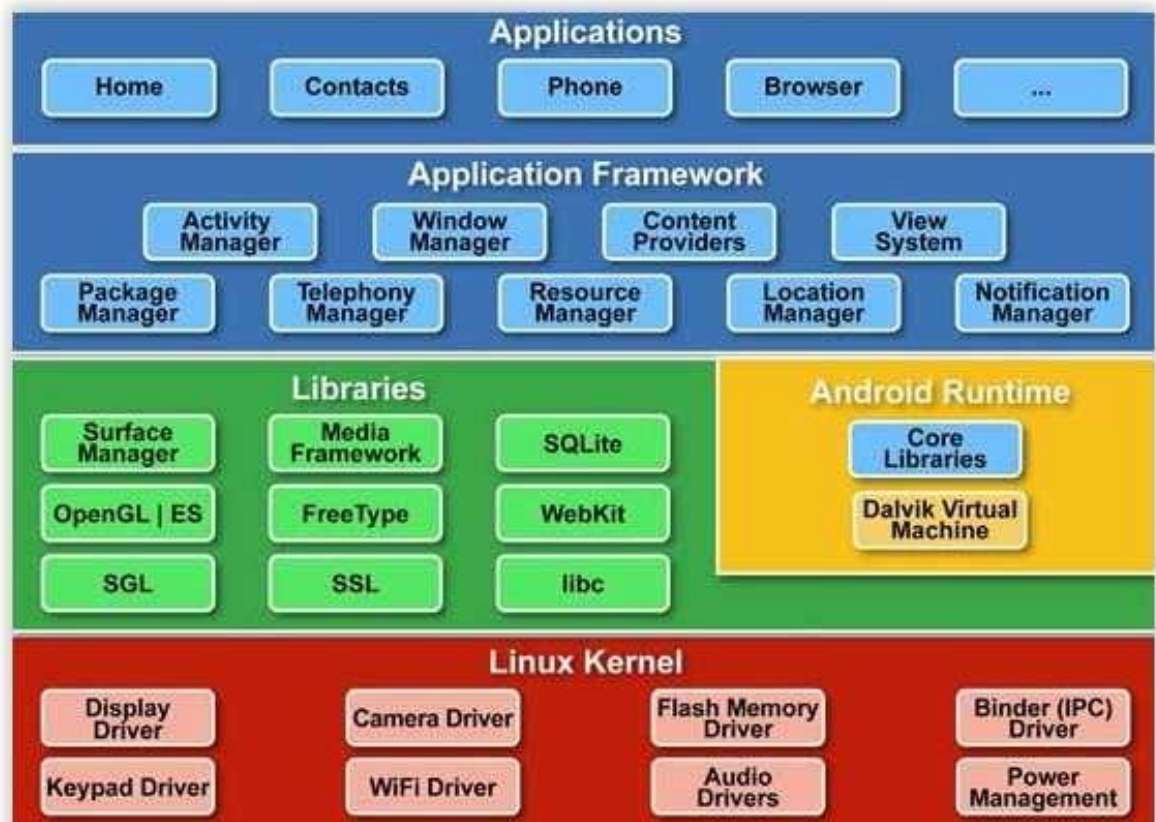
Εικόνα 17. Logo Android Q (πηγή: Android.com)

Η τρέχουσα δοκιμαστική έκδοση ανακοινώθηκε στις 13 Μαρτίου 2019, για τις συσκευές τηλεφώνου Google Pixel. Η πλήρης εμπορική λειτουργία αναμένεται το γ' τρίμηνο του 2019.

Τα νέα χαρακτηριστικά θα περιλαμβάνουν βελτιώσεις απόδοσης, καλύτερη υποστήριξη βιομετρικών δεδομένων, αύξηση επιπέδου ασφαλείας με έλεγχο πρόσβασης κλπ.

4. Η Αρχιτεκτονική του Android

Το Android είναι μια στοίβα λογισμικού η οποία περιλαμβάνει το λειτουργικό σύστημα (κατώτερο επίπεδο στοίβας), το ενδιάμεσο λογισμικό, καθώς και τις βασικές εφαρμογές (ανώτερο επίπεδο στοίβας) (Morris, 2011).



Εικόνα 18. Αρχιτεκτονική Android (πηγή: Android.com)

Η αρχιτεκτονική του λειτουργικού συστήματος Android αποτελείται από 5 βασικά επίπεδα (Hasenan, 2008).

- Τον πυρήνα Linux (Linux Kernel)
- Τις βιβλιοθήκες (Libraries)/ Την εικονική μηχανή Dalvik
- Τον χρόνο εκτέλεσης (Android Runtime)
- Το πλαίσιο εφαρμογής (Application Framework)
- Τις εφαρμογές (Application)

Αναλυτικά για τα επίπεδα αρχιτεκτονικής ισχύουν τα εξής (Gargenta, 211):

α) Πυρήνας Linux (Linux Kernel): Το λογισμικό Android έχει βασιστεί πάνω στο λειτουργικό σύστημα Linux που είναι ένα λογισμικό ανοιχτού κώδικα (open-source), ενώ είναι και συμβατό με διάφορες αρχιτεκτονικές υλικού (hardware). Σε κάθε περίπτωση το Android ανεξαρτητοποιείται από τα χαρακτηριστικά του κάθε υλικού, δηλαδή είναι συμβατό με ένα μεγάλο πλήθος συσκευών.

Το λειτουργικό Linux είναι από τα πλέον ασφαλή συστήματα, καθώς έχει δοκιμαστεί για την αξιοπιστία του σε πολύ απαιτητικά περιβάλλοντα εδώ και δεκαετίες. Η ασφάλειά που παρέχει το Linux είναι άμεσα συνδεδεμένη με την ασφάλεια του Android καθώς όλες οι εφαρμογές του τρέχουν με διαδικασίες Linux όπου οι άδειες πρόσβασης τίθενται και ελέγχονται από το ίδιο το σύστημα Linux. Επιπλέον το Android έχει ενσωματώσει πολλά χρήσιμα χαρακτηριστικά του Linux σε ζητήματα όπως η διαχείριση μνήμης, η διαχείριση ενέργειας και η δικτύωση.

β) Μητρικές Βιβλιοθήκες (Native Libraries): Συνήθως φορτώνονται από αποθετήρια της κοινότητας του ανοιχτού κώδικα/ λογισμικού και είναι γραμμένες σε γλώσσα C και C++.

Σε αυτές βρίσκονται τα απαραίτητα χαρακτηριστικά και οι ανάλογες υπηρεσίες για το επίπεδο εφαρμογών του Android (Android application layer), όπως:

- Webkit. Μια γρήγορη web-rendering μηχανή που υποστηρίζεται από πολλούς περιηγητές του διαδικτύου (browsers).
- SQLite. Μια βάση δεδομένων SQL.
- Apache Harmony. Μια open-source εφαρμογή της Java.
- OpenGL. Βιβλιοθήκες για 3D γραφικά.
- OpenSSL. Για το επίπεδο ασφαλείας lockert.

Στις περισσότερες περιπτώσεις οι βιβλιοθήκες χρησιμοποιούνται αυτούσιες εκτός από τη Bionic, η οποία χρησιμοποιείται από συσκευές με μικρής χωρητικότητας μπαταρίες, δίνοντας την ευχέρεια στο χρήστη για παραμετροποίηση των επιλογών αυτής (Meier,2010).

γ) Χρόνος εκτέλεσης Android (Android Runtime): Είναι ένα σύνολο από βιβλιοθήκες (Dalvik VM και βασικές βιβλιοθήκες/ Core libs) που προσφέρουν τη δυνατότητα ανάπτυξης εφαρμογών στη γλώσσα προγραμματισμού JAVA.

Η Dalvik VM είναι μια εικονική μηχανή (Virtual Machine) που έχει σχεδιαστεί ειδικά για το Android από τον Dan Bornstein και χρησιμοποιείται σε μια ευρεία γκάμα σκοπών και εφαρμογών. Σημαντικό χαρακτηριστικό της Dalvik VM είναι ότι η χρήση της δεν έχει οικονομικό κόστος και επιτρέπει στις συσκευές να τρέχουν τις εφαρμογές με εξοικονόμηση πόρων (χαμηλή ενέργεια και μνήμη).

Οι Core Java libraries, που προσφέρει το Android, έχουν κάποιες διαφοροποιήσεις από αυτές που προσφέρει η Java [Java Standard Edition, Java Enterprise Edition (J2EE ή JavaEE) και Java Micro Edition (J2ME ή JavaME)], με βασική διαφορά την Java user interface (AWT και Swing) την οποία το Android έχει αντικαταστήσει με αποκλειστικά δικές του βιβλιοθήκες.

δ) Πλαίσιο εφαρμογής (Application Framework): Το πλαίσιο εφαρμογής είναι ένα περιβάλλον στο οποίο περιέχεται μεγάλος αριθμός υπηρεσιών που έχουν ως σκοπό να παρέχουν βοήθεια στους προγραμματιστές για την ανάπτυξη των εφαρμογών τους.

Μέσω του πλαισίου εφαρμογής υπάρχει πρόσβαση σε μεγάλο αριθμό βιβλιοθηκών Java, που αποτελούνται από πολλές υπηρεσίες που δίνουν τη δυνατότητα αλληλεπίδρασης και διαχείρισης των εφαρμογών με τις βασικές λειτουργίες της συσκευής όπως η τηλεφωνία, το WiFi, η τοποθεσία κλπ, αποκλειστικά για το Android.

ε) Εφαρμογές (Applications): Βρίσκονται στο ανώτερο επίπεδο της αρχιτεκτονικής Android, και είναι στην ουσία οι εφαρμογές που δημιουργούνται από τις εταιρείες και από τους developers. Οι εφαρμογές αυτές συνοδεύονται από λειτουργίες και χαρακτηριστικά που είναι χρήσιμα στον χρήστη και βρίσκονται συνήθως προεγκατεστημένες στη συσκευή ή ο χρήστης μπορεί να τις προμηθευτεί από το Playstore.

Κάθε τέτοια εφαρμογή αποτελείται από ένα αρχείο και το μοναδικό πακέτο εφαρμογής (single application package- APK), το οποίο αποτελείται από τρία βασικά στοιχεία: Dalvik executable, Resources και Native libraries.

5. Δομικά χαρακτηριστικά εφαρμογών

Μια εφαρμογή Android περιέχει ένα ή περισσότερα από τα παρακάτω χαρακτηριστικά (Gargenta, 2011) :

α) Δραστηριότητες (Activities). Περιέχουν εκτελέσιμο κώδικα του οποίου η λειτουργικότητα μπορεί να ενεργοποιηθεί είτε από τον χρήστη είτε από το ίδιο το λειτουργικό σύστημα και μπορούν να επικοινωνούν μεταξύ τους και να ανταλλάσσουν πληροφορίες.

β) Υπηρεσίες (Services). Οι υπηρεσίες αν και περιέχουν εκτελέσιμο κώδικα η λειτουργικότητα που παρέχουν δεν περιλαμβάνει γραφικό περιβάλλον αλληλεπίδρασης με τον χρήστη (user interface). Συνεπώς, οι λειτουργικότητες που προκύπτουν από τις υπηρεσίες δεν απαιτούν άμεση αλληλεπίδραση με τον χρήστη και μπορούν να εκτελούνται παράλληλα με κάποια άλλη δραστηριότητα, ενώ μπορούν να επικοινωνούν ταυτόχρονα με άλλα στοιχεία του Android.

γ) Δέκτες εκπομπής (Broadcast Receiver). Είναι τα στοιχεία που ανταποκρίνονται στις εκπομπές των αντίστοιχων σημάτων που μεταδίδονται σε όλες τις εφαρμογές του λειτουργικού συστήματος. Τα σήματα αυτά παράγονται είτε από το ίδιο το λειτουργικό σύστημα είτε από κάποια άλλη εφαρμογή με σκοπό την ενημέρωση του χρήστη για ένα γεγονός έξω από την εφαρμογή, π.χ. σύνδεση Bluetooth.

δ) Πάροχοι περιεχομένου (content provider). Οι πάροχοι ορίζουν τη σχέση των εφαρμογών με τα δεδομένα της, δηλαδή διαμερισμό και αποθήκευση δεδομένων σε βάσεις δεδομένων SQLite και οποιαδήποτε άλλη δυνατή μορφή.

ε) Intents. Αποτελούν το κύριο χαρακτηριστικό για την επικοινωνία μεταξύ των διαφορετικών δραστηριοτήτων της εφαρμογής για την ανταλλαγή πληροφοριών και δεδομένων μεταξύ τους. Η χρήση των intents μας δίνει τη δυνατότητα να εναλλάσσουμε δραστηριότητες και επιπλέον παρέχει τη δυνατότητα σε μια εφαρμογή να επιλέξει μια δραστηριότητα βασιζόμενη στην ενέργεια που θέλει να εκτελέσει και των διαθέσιμων δεδομένων.

6. Εσωτερικό Εφαρμογής Android

Το εσωτερικό μιας εφαρμογής Android, όπου είναι δομημένα μια σειρά αρχείων, αποτελείται από τους εξής φακέλους (Steele , 2010):

- AndroidManifest.xml, μέσα σε αυτό το xml αρχείο είναι καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα, όπως το όνομα πακέτου της εφαρμογής, οι άδειες χρήσης της εφαρμογής, όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.
- Φάκελος src, μέσα στο φάκελο αυτό περιέχονται όλα τα αρχεία κλάσης της Java όλων των Activities, Services κλπ. και ταυτόχρονα αποτελεί τον μοναδικό φάκελο στον οποίο αποθηκεύονται τα αρχεία του κώδικα.

- Φάκελος `res`, περιέχει όλα τα αρχεία εικόνας, κειμένου, `layout`, κλπ τα οποία χρησιμοποιούνται από τις `Activities` που βρίσκονται στον φάκελο `src` ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους.

7. Ασφάλεια στο Android

Κάθε εφαρμογή που εγκαθίσταται στη συσκευή λειτουργεί στη δική της εικονική μηχανή (`Virtual machine`) και αυτό είναι πλαίσιο ασφαλείας της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα το οποίο αντιμετωπίζει την κάθε εφαρμογή σαν ξεχωριστό χρήστη.

Το σύστημα παρέχει ένα μοναδικό αριθμό ID ο οποίος είναι άγνωστος στην εφαρμογή και καθώς αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της κάθε εφαρμογής, μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.

Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή απομονωμένη από τις υπόλοιπες εφαρμογές μόλις της 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 (Singh, 2014).

ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ

1. Εργαλεία ανάπτυξης λογισμικού για Android

Για να βοηθήσει τους developers το Android παρέχει το Android Software development kit (SDK), το οποίο προσφέρει τη γλώσσα προγραμματισμού Java για την ανάπτυξη των εφαρμογών, καθώς και μια σειρά από άλλα χαρακτηριστικά όπως debugger, βιβλιοθήκες, προσομοιωτές, παραδείγματα κώδικα, οδηγίες κλπ.

Το Android SDK (Software Development Kit) είναι ένα σύνολο από βασικά εργαλεία που μπορούν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών για Android, όπως εργαλεία για εκσφαλμάτωση του προγράμματος (debugger), βιβλιοθήκες (libraries), εργαλείο προσομοίωσης QEMU (emulator), αρχεία τεκμηρίωσης, δείγματα κώδικα και βοηθήματα.

Στις πλατφόρμες που υποστηρίζονται αυτή τη στιγμή περιλαμβάνονται το Linux, Mac OS X 10.5.8 ή μεταγενέστερο και Windows 7 ή μεταγενέστερα. Σε κάθε έκδοση του SDK αντιστοιχεί ένας μοναδικός αριθμός που εκφράζει το API Level της έκδοσης. Κατά την ανάπτυξη ενός προγράμματος Android θα πρέπει να καθοριστεί η ελάχιστη έκδοση SDK στην οποία μπορεί να εκτελεστεί το πρόγραμμα και αναφέρεται ως minSdkVersion. Ως targetSdkVersion καθορίζεται η έκδοση όπου η εφαρμογή έχει κυρίως δοκιμαστεί. Τέλος, με compileSdkVersion αναφέρεται εκείνη η έκδοση του SDK στην οποία γίνεται η μεταγλώττιση του προγράμματος.

Οι βιβλιοθήκες που έχουν γραφτεί σε γλώσσα C/C++ μπορούν να μεταγλωττιστούν και να εκτελεστούν από ένα σύστημα Android με τη χρήση του Android NDK (Native Development Kit). Οι βιβλιοθήκες αυτές μπορούν να 'τρέξουν' με τη βοήθεια της κλήσης System.loadLibrary του Dalvik VM, οπότε έτσι έχουμε εφαρμογές που εκτελούνται σε περιβάλλον Android αλλά έχουν αναπτυχθεί σε C/C++ με τη χρήση άλλων παραδοσιακών εργαλείων.

Μερικά από τα εργαλεία τα οποία μπορούν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών Android, εμφανίζονται στην πιο κάτω λίστα (Lee, 2011):

- Eclipse. Είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) το οποίο μέχρι το 2014 αποτελούσε το επίσημο υποστηριζόμενο

περιβάλλον για την ανάπτυξη εφαρμογών Android. Απαιτεί την εγκατάσταση του προσθέτου ADT (Android Development Tools) για την υποστήριξη προγραμματισμού Android.

- Android Studio. Είναι IDE περιβάλλον που από το 2015 αποτελεί πλέον το επίσημα υποστηριζόμενο από την Google για την ανάπτυξη εφαρμογών Android και βασίζεται στο περιβάλλον IntelliJ IDEA της JetBrains, που είναι ένα εργαλείο ανάπτυξης προγραμμάτων Java.
- App Inventor. Είναι online εργαλείο στο οποίο μπορεί να γίνει η ανάπτυξη μίας Android εφαρμογής σε γραφικό περιβάλλον. Αναπτύχθηκε από τη Google το 2010 και βασίστηκε στη βιβλιοθήκη Open Blocks Java, ενώ προτιμάται από αρχάριους προγραμματιστές.
- Library του MIT. Είναι κώδικας ελεύθερα διαθέσιμος με ευθύνη της ανάπτυξής του από το MIT.
- Apache Cordova. Γνωστό και ως PhoneGap, επιτρέπει τη δημιουργία εφαρμογών για φορητές συσκευές με τη χρήση των γλωσσών CSS3, HTML5 και Javascript ενώ οι εφαρμογές του μπορούν να εκτελεστούν σε διαφορετικές πλατφόρμες, τόσο σε Android όσο και σε iOS.
- Xamarin. Είναι πλατφόρμα βασισμένη στη C++ που επιτρέπει την ανάπτυξη εφαρμογών για Android και iOS, ενώ μπορούν να χρησιμοποιηθούν βιβλιοθήκες .NET κατά την μεταγλώττιση του κώδικα.
- Appcelerator. Θεωρείται από τα φιλικότερα περιβάλλοντα ανάπτυξης, επιτρέποντας την ανάπτυξη εφαρμογών κάνοντας χρήση της γλώσσας προγραμματισμού Javascript, με τις εφαρμογές να είναι συμβατές τόσο για Android, όσο και για iOS.

2. Γενικά το Android Studio

Το Android Studio αποτελεί ένα προγραμματιστικό περιβάλλον για την ανάπτυξη εφαρμογών της πλατφόρμας Android, που κυκλοφόρησε τον Δεκέμβριο του 2014 και είναι διαθέσιμο για λειτουργικά συστήματα Windows, Mac OS X και Linux.

Όπως έχει προαναφερθεί το Android Studio είναι ένα integrated development environment (IDE) το οποίο δημιουργήθηκε από την Google, βασιζόμενη στο IntelliJ IDE

Ορισμένα βασικά χαρακτηριστικά και λειτουργίες που παρέχει είναι:

- Flexible Gradle-based build system.
- Προσομοιωτή.
- Περιβάλλον ανάπτυξης εφαρμογών για όλες τις συσκευές που υποστηρίζουν Android.
- Λειτουργία Instant Run για να μεταδίδονται οι αλλαγές που γίνονται στην εφαρμογή χωρίς να δημιουργηθεί καινούργιο APK.
- Δείγματα κώδικα και υποστήριξη του GitHub integration με σκοπό την εισαγωγή ή το ανέβασμα κώδικα από το GitHub.
- Υποστήριξη του NDK και της γλώσσας προγραμματισμού C++.

3. Android software development kit (SDK)

Το Android software development kit (SDK) περιέχει πρόσθετες βιβλιοθήκες, εργαλεία ανάπτυξης καθώς και έναν προσομοιωτή που απαιτείται για τη δοκιμή της και επίσης συμπεριλαμβάνει την εικονική μηχανή Dalvik η οποία εκτελεί τον κώδικα για τις ανάγκες του λειτουργικού συστήματος Android.

Μερικά πρόσθετα του SDK είναι το Android Development Tools (ADT) που περιέχει εργαλεία για τον σχεδιασμό των γραφικών στοιχείων και άλλων πρόσθετων λειτουργιών που διευκολύνουν τη συγγραφή κώδικα όπως:

- το περιβάλλον διαχείρισης εικονικών μηχανών προσομοίωσης των εφαρμογών (Android Virtual Devices Manager -AVD Manager),
- τον προσομοιωτή συσκευής (Android Emulator) για τη δοκιμή των εφαρμογών
- τον διαχειριστή στοιχείων του SDK (SDK Manager) όπου μέσω του γραφικού περιβάλλοντος επιτρέπεται η διαχείριση πακέτων και βιβλιοθηκών.

4. Χαρακτηριστικά του Android Studio IDE

Το Android Studio έχει τα εξής χαρακτηριστικά:

- Ισχυρή Επεξεργασία Κώδικα (Powerful Code Editing) με έξυπνο σύστημα αυτοσυμπλήρωσης κώδικα, ριζική πλοήγηση μεταξύ των αρχείων του έργου (Profound navigation between project files), προχωρημένη και ασφαλή επεξεργασία (π.χ. μετονομασία ενός αρχείου και εφαρμογή του νέου ονόματος σε όλα τα μέρη του κώδικα που αναφέρεται αυτόματα) και προεπισκόπηση πόρων.
- Λειτουργία ανάλυσης κώδικα κατά την επεξεργασία (On-the-fly Code Analysis), κατά την οποία ο συντάκτης επισημαίνει προειδοποιήσεις και σφάλματα στον κώδικα αμέσως καθώς πληκτρολογείται και επιτρέπει να εφαρμοστεί μια προτεινόμενη γρήγορη λύση αυτόματα.
- Ενσωματωμένα εργαλεία του Android (Built-in Android Tools) που περιλαμβάνουν:
 - Ισχυρό σχεδιαστή διεπαφής χρήστη με drag-n-drop και υποστήριξη διαφορετικών διατάξεων και μεγεθών οθόνης.
 - Ενσωμάτωση του συστήματος καταγραφής LogCat με δυνατότητα αναζήτησης και εφαρμογής διάφορων φίλτρων εμφάνισης.
 - Ενσωμάτωση του εργαλείου αποσφαλμάτωσης Dalvik Debug Monitor Server (DDMS).
 - Λειτουργία θέασης ιεραρχίας (Hierarchy View) που δίνει τη δυνατότητα να δούμε την ιεραρχία των στοιχείων της διεπαφής χρήστη.

- Περιέχει εκτέλεση και αποσφαλμάτωση (debugger) για εξομοιώσεις κινητών συσκευών καθώς και πραγματικών συσκευών.

- Μέσω του εργαλείου Draw 9 προσφέρει την επεξεργασία-δημιουργία εικόνων Bitmap και την αυτόματη προσαρμογή τους σε κάθε μεγέθους οθόνη που τις φιλοξενεί.

Άλλα χαρακτηριστικά

- Το ενοποιημένο σύστημα Proguard που δίνει την δυνατότητα τροποποιήσεων του κώδικα, έχοντας ως αποτέλεσμα ένα μικρότερου μεγέθους αρχείο .apk, πιο δύσκολο να αντιστραφεί μηχανικά (Reverse Engineered) με συνέπεια την αποτροπή υποκλοπής του κώδικα.

- Το Android απαιτεί από τις εφαρμογές να είναι ψηφιακά υπογεγραμμένες με ένα πιστοποιητικό για να μπορέσουν να εγκατασταθούν. Παρέχεται μέσω του Android Studio η δυνατότητα ψηφιακής υπογραφής της παραχθείσας εφαρμογής με πιστοποιητικό (Certificate) για την ταυτοποίηση του συγγραφέα/προγραμματιστή ενός app, χωρίς να χρειάζεται να υπογραφεί από κάποια αρχή έκδοσης πιστοποιητικών. Συνήθως, οι Android εφαρμογές χρησιμοποιούν τα λεγόμενα αυτό-υπογεγραμμένα πιστοποιητικά όπου ο προγραμματιστής της εφαρμογής κατέχει το ιδιωτικό κλειδί του πιστοποιητικού.

- Υπάρχει ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud, καθιστώντας εύκολη την ενσωμάτωση του Google Cloud Messaging και του App Engine.

- Η εργαλειοθήκη Lint (Lint tools) που(ίσως χωρίς το “που”; δεν έχεις κύρια πρόταση εδώ) δίνει τη δυνατότητα καταγραφής της απόδοσης, της χρησιμότητας, της συμβατότητας εκδόσεων του Android, και άλλων προβλημάτων του πηγαίου κώδικα που μπορούν να προκαλέσουν σφάλματα.

- Πρότυπα κώδικα (Code templates) που βοηθούν να αναπτυχθούν εφαρμογές κοινών χαρακτηριστικών.

5. Εργαλεία ανάπτυξης λογισμικού Android (Android Software Development Kit SDK)

Εργαλεία Εικονικών Συσκευών (Android Virtual Device Tools AVD)

- Διαχειριστής Εικονικών Συσκευών (Android Virtual Device Manager).
- Εξομοιωτής των Android συσκευών (Android Emulator).
- Εργαλείο δημιουργίας εξωτερικής μνήμης για τις εικονικές συσκευές Android (mkcard) πχ. Εξωτερική κάρτα μνήμης (SD card).

Εργαλεία ανάπτυξης (Development Tools)

- Λειτουργία θέασης ιεραρχίας (Hierarchy View)
- Εργαλείο Lint (Lint tools)
- Ο Διαχειριστής Εργαλείων Ανάπτυξης Εφαρμογών (SDK Manager).
- Το `sqlite3`, προσφέρει τη διαχείριση βάσεων δεδομένων (Databases) που δημιουργήθηκαν από εφαρμογές του Android, ενώ περιλαμβάνει πολλές χρήσιμες εντολές δίνοντας επίσης τη δυνατότητα για εκτέλεση εντολών SQLite κατά την επεξεργασία.

Εργαλεία αποσφαλμάτωσης (Debugging Tools)

- Android Debug Bridge ADB, ένα ευέλικτο εργαλείο γραμμής εντολών που επιτρέπει την επικοινωνία με ενεργό εξομοιωτή ή συνδεδεμένη συσκευή Android. Παρέχει επίσης πρόσβαση στο κέλυφος της συσκευής (Device Shell), προσφέροντας πληθώρα εντολών για την αποσφαλμάτωση.
- Εργαλείο αποσφαλμάτωσης Dalvik Debug Monitor Server (DDMS).

- Εργαλείο παρακολούθησης συσκευής (Android Device Monitor) με δυνατότητες αποσφαλμάτωσης και ανάλυσης χρήσης πόρων του συστήματος (κύριας μνήμης, κεντρικής μονάδας επεξεργασίας ΚΜΕ κ.α.).
- Dmtracedump (Trace Dump Diagram). Δημιουργεί τα γραφικά διαγράμματα της στοίβας κλήσεων (Call Stack) από τα αρχεία καταγραφής παρακολούθησης (Log Files) και χρησιμοποιεί το βοηθητικό λογισμικό Graphviz Dot για να δημιουργήσει τη γραφική έξοδο.
- Traceview, παρόμοιο εργαλείο με το Dmtracedump.
- SysTrace (System Trace). Το εργαλείο Systrace βοηθά στην ανάλυση της απόδοσης της εφαρμογής μέσω της σύλληψης και της προβολής του χρόνου εκτέλεσης των διεργασιών των εφαρμογών και άλλων διεργασιών του λειτουργικού συστήματος. Το εργαλείο συνδυάζει δεδομένα από τον πυρήνα του Android, όπως το χρονοδιάγραμμα της CPU, τη δραστηριότητα του δίσκου, και τα νήματα της εφαρμογής (Application Threads) για να δημιουργήσει μια HTML έκθεση που παρουσιάζει μια συνολική εικόνα των διεργασιών του συστήματος της συσκευής Android για μια δεδομένη χρονική περίοδο.

Εργαλεία Κατασκευής (Build Tools)

- Java Opaque Binary Blob (JOBB). Το JOBB επιτρέπει τη δημιουργία κρυπτογραφημένων και μη κρυπτογραφημένων αρχείων επέκτασης .apk σε δυαδική μορφή (Opaque Binary Blob), που χρησιμοποιούνται για την παροχή πρόσθετων στοιχείων του αρχείου .apk (όπως γραφικά, ήχους και βίντεο).
- ProGuard
- Zipalign (.zip align). Η χρήση του βελτιστοποιεί τα .apk αρχεία καθώς τα δεδομένα ξεκινούν με μια ιδιαίτερη ευθυγράμμιση σχετική με την αρχή του αρχείου.

Εργαλεία Επεξεργασίας Εικόνων (Image Tools)

- Draw 9
- Vector Asset Studio. Το Vector Asset Studio βοηθάει στην προσθήκη εικονιδίων υλικού (Material Icons) και στην εισαγωγή Scalable Vector Graphic (SVG). Σε σύγκριση με τις εικόνες raster, διανυσματικά drawables μπορούν να μειώσουν το μέγεθος της εφαρμογής αλλάζοντας το μέγεθος της εικόνας χωρίς απώλεια.
- etc1tool (Etc1 Compression Tool). Είναι βοηθητικό πρόγραμμα γραμμής εντολών που επιτρέπει την κωδικοποίηση PNG εικόνων από το πρότυπο συμπίεσης ETC1 και το αντίστροφο.
- Tracer for OpenGL ES. Εντοπίζει τις εντολές OpenGL ES (Embedded Systems) παρέχοντας την σταδιακή ανάλυση των εντολών που αφορά στα γραφικά.

Εργαλεία Πλατφόρμας (Platform Tools)

- Android Debug Bridge (adb). Είναι εργαλείο γραμμής εντολών που επιτρέπει την επικοινωνία με ένα στιγμιότυπο εξομοιωτή ή μια συνδεδεμένη συσκευή Android, τον έλεγχο της κατάστασής τους και την εγκατάσταση μιας εφαρμογής Android (.apk).
- Bmgr (Back Up Manager). Είναι εργαλείο κελύφους που χρησιμοποιείται για αλληλεπίδραση με τον Διαχειριστή αντιγράφων ασφαλείας (Back-up Manager) σε συσκευές Android που υποστηρίζουν το API Level 8 (FROYO) ή μεγαλύτερο.
- LogCat. Είναι εργαλείο με δυνατότητα αναζήτησης και εφαρμογής διάφορων φίλτρων εμφάνισης.

Πακέτα που περιλαμβάνονται στο Android SDK

- SDK Tools

- SDK Platform-tools
- SDK Platforms. Μία πλατφόρμα απαιτείται να υπάρχει στο περιβάλλον για να γίνει μεταγλώττιση της εφαρμογής.
- System Images. Περιλαμβάνουν τις εικόνες των συστημάτων των συσκευών για όλες τις εκδόσεις του Android που υποστηρίζει η εφαρμογή και επιτρέπουν να γίνει δοκιμή της εφαρμογής με τον εξομοιωτή συσκευών Android (Android emulator).
- Android Support Library. Περιλαμβάνει μια στατική βιβλιοθήκη που επιτρέπει τη χρήση των Android API (όπως fragments, καθώς και άλλα που δεν περιλαμβάνονται στο framework καθόλου) για συσκευές που τρέχουν παλαιότερη έκδοση πλατφόρμας. Όλα τα πρότυπα δραστηριοτήτων (activity templates) που είναι διαθέσιμα απαιτούν την ύπαρξη του ASL όταν δημιουργείται ένα νέο έργο (new project) με το πρόσθετο ADT (ADT Plugin).
- Android Support Repository. Περιλαμβάνει το τοπικό αποθετήριο Maven (local Maven repository) για τις βιβλιοθήκες υποστήριξης.
- Google Play services. Περιλαμβάνει τις βιβλιοθήκες υπηρεσιών Google Play για την πλευρά του πελάτη (Google Play services client library) καθώς και κάποια δείγματα κώδικα (sample code).
- Google Repository. Περιλαμβάνει το τοπικό αποθετήριο Maven (local Maven repository) για τις βιβλιοθήκες της Google (Google libraries).

6. Ανάπτυξη εφαρμογής MyHUApp



Εικόνα 19. Κεντρικό μενού εφαρμογής

Η εφαρμογή έχει ονομαστεί δοκιμαστικά ως myHUApp και το κεντρικό της μενού αποτελείται από 6 πεδία επιλογής (εικόνα 19) :

- Γενικές Πληροφορίες
- Τμήματα
- Υπηρεσίες
- Επικοινωνία

- SOS
- Πλοήγηση



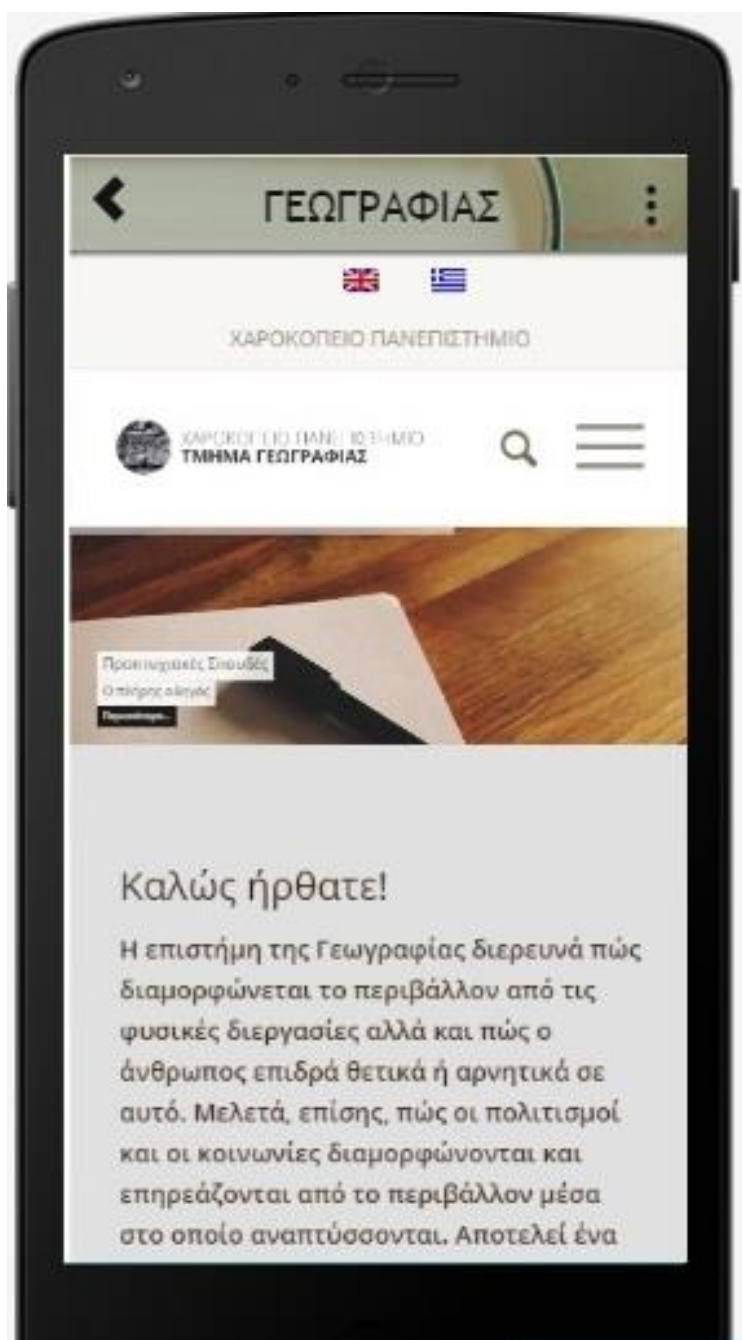
Εικόνα 20. Επιλογή 'Γενικών Πληροφοριών'

Επιλέγοντας τις 'Γενικές Πληροφορίες' έχουμε πρόσβαση στο ιστορικό του ιδρύματος, όπως αυτό έχει δημοσιευτεί στην επίσημη ιστοσελίδα του πανεπιστημίου (εικόνα 20).



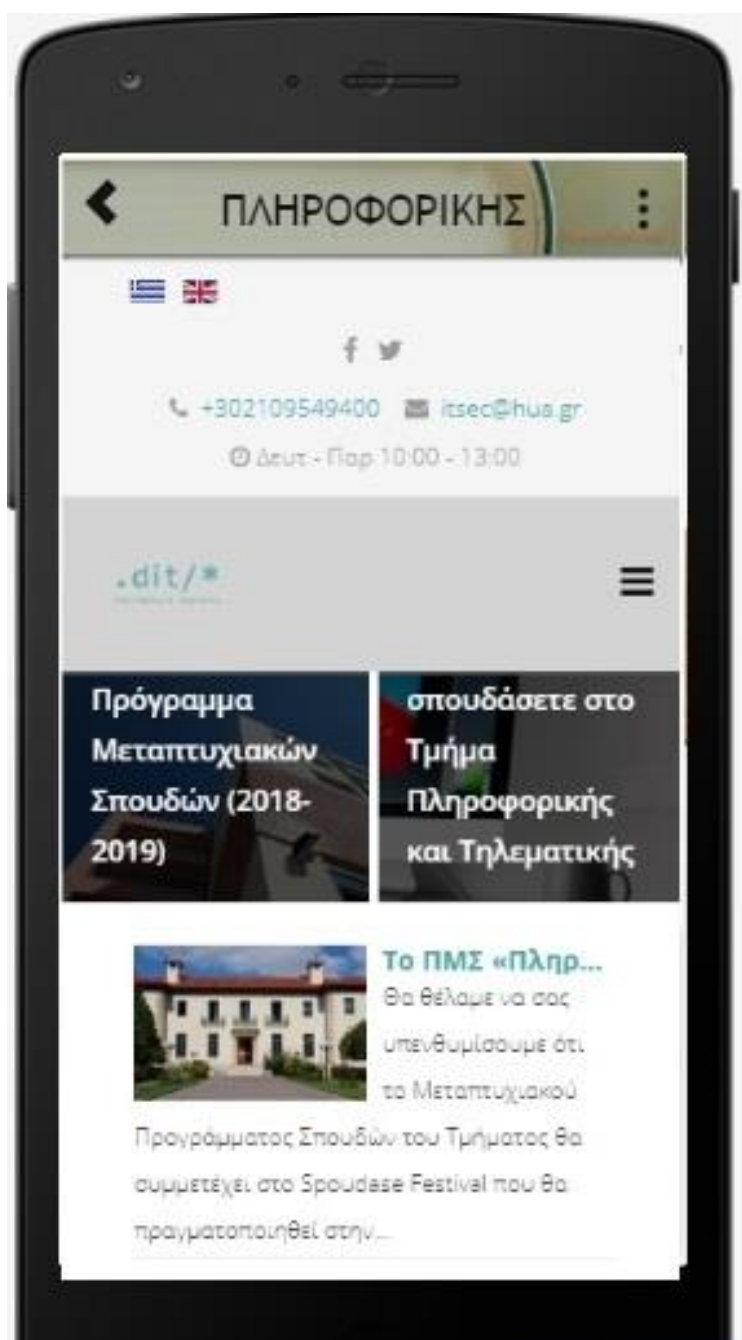
Εικόνα 21. Μενού 'Τμημάτων'

Επιλέγοντας το πεδίο 'Τμήματα' εμφανίζονται σε δεύτερο επίπεδο τα τέσσερα τμήματα του πανεπιστημίου (εικόνα 21). Επειδή δεν υπάρχουν ξεχωριστά εγκεκριμένα λογότυπα για το κάθε τμήμα επιλέχθηκαν κάποια αντιπροσωπευτικά σχέδια από ελεύθερη ηλεκτρονική βιβλιοθήκη.



Εικόνα 22. Πλοήγηση επιλογής 'Γεωγραφίας'

Επιλέγοντας το εικονίδιο κάθε τμήματος μπορούμε να γίνει πλοήγηση εστην αντίστοιχη ιστοσελίδα (εικόνες 22-23-24-25) έχοντας πρόσβαση σε επιπλέον πληροφορίες. Σημειώνεται πως η ιστοσελίδα που είχε πλήρη συμβατότητα για προβολή σε συσκευή κινητού τηλεφώνου ήταν αυτή του τμήματος Πληροφορικής- Τηλεματικής.



Εικόνα 23. Πλοήγηση επιλογής 'Πληροφορικής'



Εικόνα 24. Πλοήγηση επιλογής 'Διαιτολογίας'



Εικόνα 25. Πλοήγηση επιλογής 'Οικ. Οικονομίας'



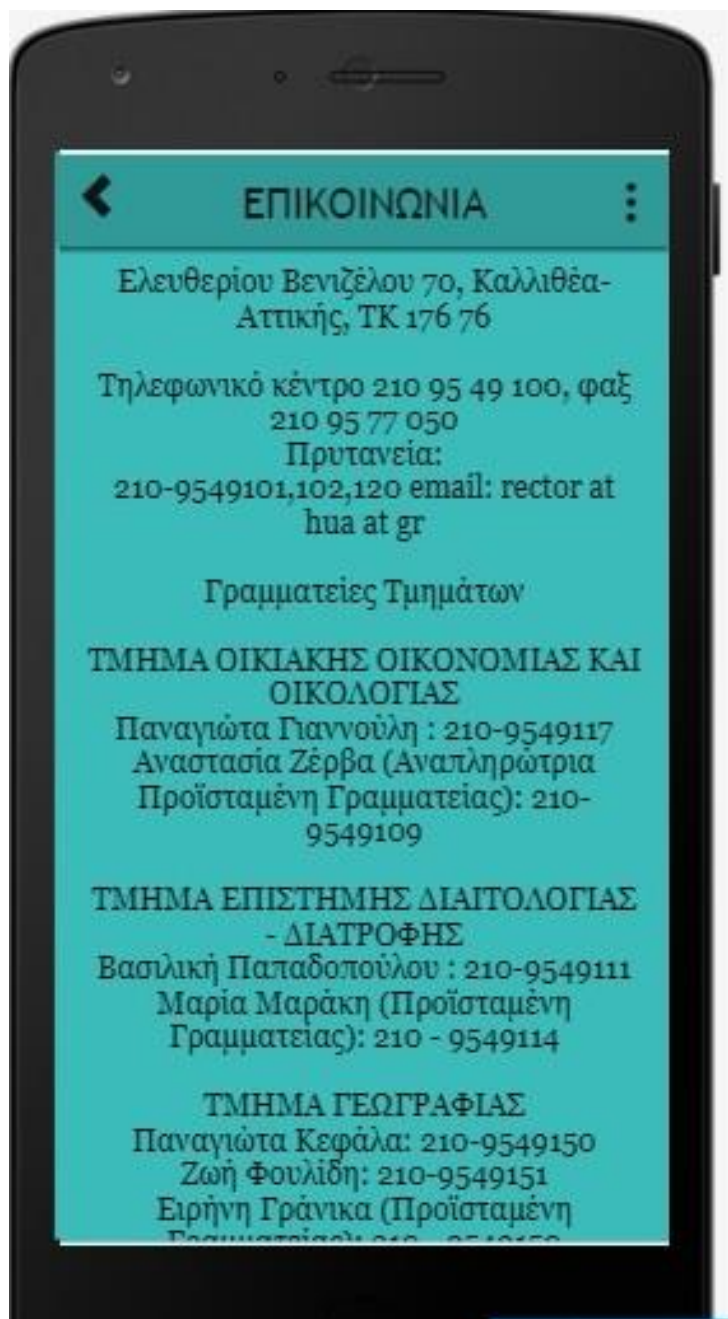
Εικόνα 26. Μενού 'Υπηρεσιών'

Επιλέγοντας στην αρχική οθόνη της εφαρμογής την επιλογή 'Υπηρεσίες' , δίνεται πρόσβαση στα εξής (εικόνα 26):

- Webmail
- E-class
- Ηλεκτρονικές Υπηρεσίες HUA

- Βιβλιοθήκη & Κέντρο Πληροφόρησης

Η πρόσβαση στις συγκεκριμένες υπηρεσίες γίνεται μέσω νέου παραθύρου σε web-browser λόγω περιορισμού πρόσβασης, ελέγχου δικαιωμάτων και διαδικασίας σύνδεσης (log-in).



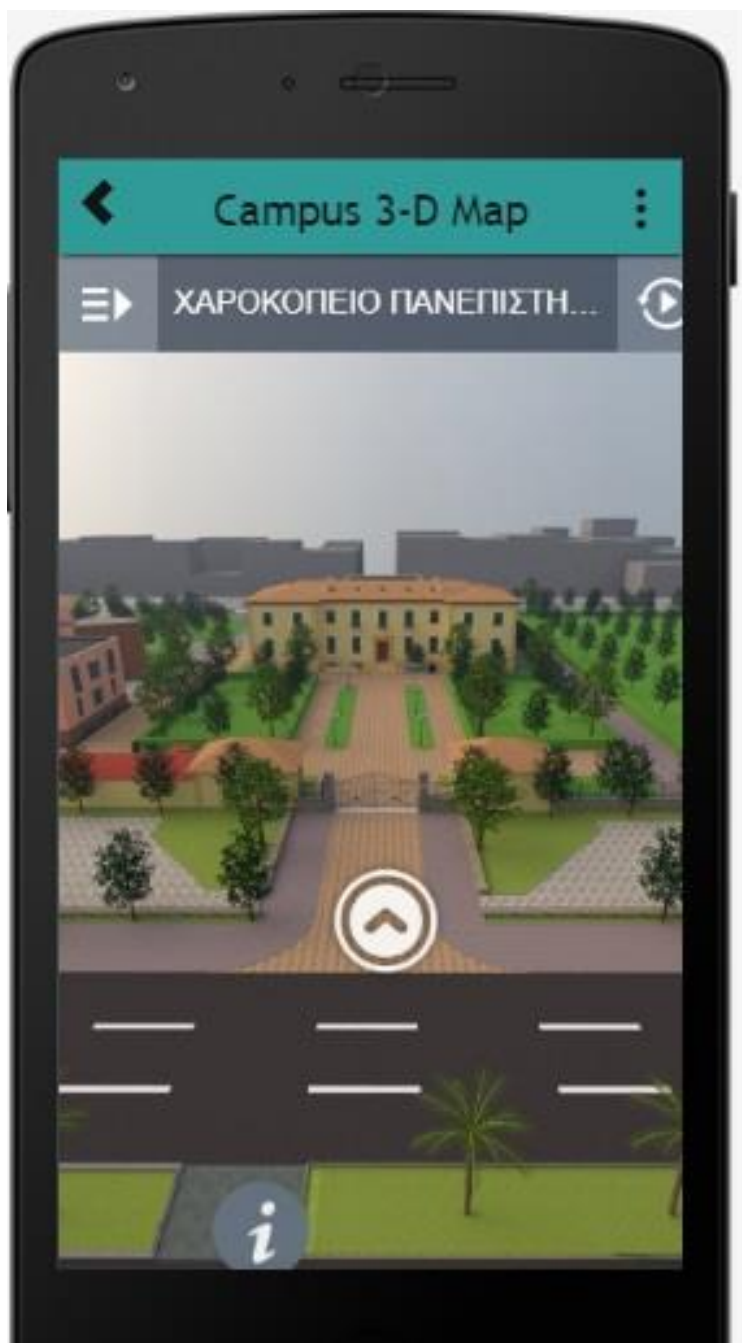
Εικόνα 27. Πλοήγηση επιλογής 'Επικοινωνίας'

Επιλέγοντας από την αρχική οθόνη 'Επικοινωνία' παρουσιάζεται η πλήρης εικόνα των στοιχείων επικοινωνίας, όπως αυτά αποτυπώνονται στην ιστοσελίδα του πανεπιστημίου (εικόνα 27) . Σε αυτό το επίπεδο, υπάρχει η δυνατότητα επιλογής αριθμού για άμεση κλήση από το κινητό.

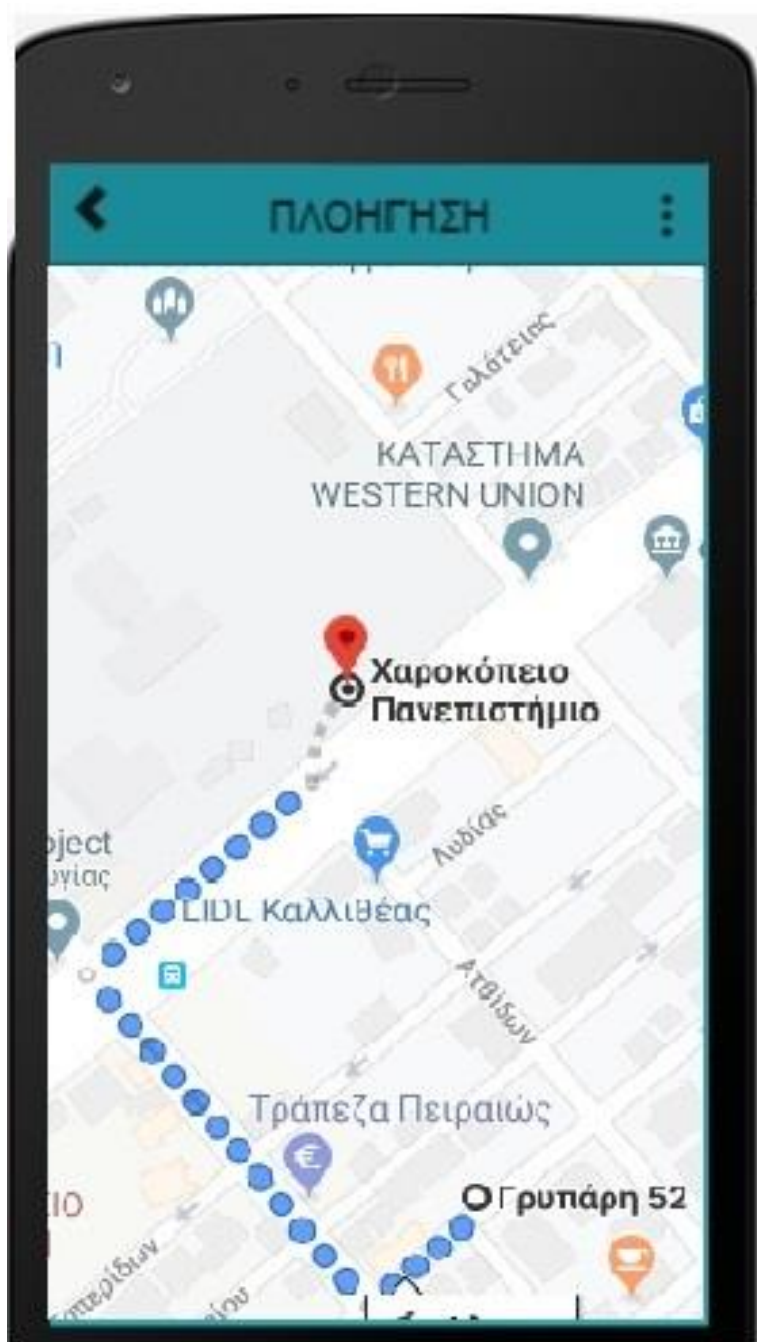


Εικόνα 28. Μενού 'Πλοήγηση'

Η επιλογή 'Πλοήγηση' μεταφέρει το χρήστη σε επόμενο μενού (εικόνα 28), όπου και μπορεί να επιλέξει είτε την εμφάνιση του τρισδιάστατου χάρτη του πανεπιστημίου (εικόνα 29), είτε δίνοντας άδεια χρήσης του εντοπισμού της θέσης να γίνει σύνδεση με την πλατφόρμα Google Maps από όπου θα λάβει πληροφορίες για τη μετακίνηση από/προς το Χαροκόπειο (εικόνα 30).



Εικόνα 29. Πλοήγηση επιλογής 'Campus 3-D Map'



Εικόνα 30. Πλοήγηση επιλογής 'Take me to HUA'



Εικόνα 31. Πλοήγηση επιλογής '112-SOS'

Τέλος, δίνοντας πάλι τις κατάλληλες άδειες χρήσης στην εφαρμογή, υπάρχει η επιλογή άμεσης κλήσης στον πανευρωπαϊκό αριθμό άμεσης ανάγκης 112 (εικόνα 31), σημειώνοντας τα εξής :

- είναι χωρίς χρέωση και μπορεί να γίνει κλήση από κινητό τηλέφωνο ακόμη και χωρίς κάρτα SIM
- λειτουργεί εντός της εμβέλειας οποιουδήποτε δικτύου κινητής τηλεφωνίας. Αν η περιοχή του συμβάντος δεν καλύπτεται από το δίκτυο κινητής τηλεφωνίας του καλούντος, η κλήση προς το 112 πραγματοποιείται μέσω άλλων δικτύων κινητής τηλεφωνίας που καλύπτουν τη συγκεκριμένη περιοχή
- παρέχει τη δυνατότητα εντοπισμού της θέσης του καλούντος
- λειτουργεί σε όλα τα κράτη μέλη της Ε.Ε παράλληλα με τους εθνικούς αριθμούς κλήσης έκτακτης ανάγκης.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ανάπτυξη μιας εφαρμογής Android είναι μια ιδιαίτερη εργασία, η οποία μέχρι πρότινος απαιτούσε μεγάλη εμπειρία στον προγραμματισμό. Πλέον παρέχεται μέσω μιας σειράς εργαλείων και λογισμικών, σε κάθε άτομο με μικρή προγραμματιστική εμπειρία η δυνατότητα ανάπτυξης εφαρμογής.

Η αναζήτηση λύσεων κατά το σχεδιασμό και την υλοποίηση του προγράμματος οδήγησαν στην απόκτηση νέων δεξιοτήτων. Αυτές σε συνδυασμό με τη χρήση διαφορετικών τεχνολογιών και λογισμικών αποτελούν πολύτιμη τεχνογνωσία για μελλοντικές προσπάθειες.

Τηρουμένων των αναλογιών, η εφαρμογή που αναπτύχθηκε προσφέρει μια ολοκληρωμένη παρουσίαση του Χαροκόπειου Πανεπιστημίου και παρέχει βασικές ενημερωτικές υπηρεσίες προς τους χρήστες.

Η προσπάθεια και οι γνώσεις που μένουν στο τελείωμα αυτής της προσπάθειας αποτελούν οδηγό για μελλοντικές προκλήσεις σε όχι και τόσο οικεία γνωστικά αντικείμενα, αντικείμενα όμως που βρίσκονται σε ιδιαίτερη ζήτηση τόσο σε ακαδημαϊκό όσο και σε επαγγελματικό επίπεδο.

Σχέδια βελτίωσης- εξέλιξης της εφαρμογής

- Βελτίωση interface της εφαρμογής, προσθέτοντας καλύτερες εικόνες ή ακόμα και 3D γραφικά.
- Σύνδεση με φοιτητολόγιο, γραφείο πρακτικής κλπ.
- Δυνατότητα μεταφοράς της εφαρμογής στην εξωτερική μονάδα αποθήκευσης (sd card).
- Δημιουργία προσωπικού προφίλ (τμήμα, μαθήματα, καθηγητές κλπ) μέσα στην εφαρμογή για καλύτερη εμπειρία χρήστη.
- Υλοποίηση εφαρμογής και για άλλα λειτουργικά κινητών (iOS, Window mobile κλπ)
- Ενσωμάτωση υπηρεσίας ειδοποίησης (notification) για νέες ανακοινώσεις, υπενθύμιση μαθημάτων, εργασιών, εξετάσεων κλπ.
- Εμπλουτισμό με επιπλέον χρήσιμες πληροφορίες (κέντρο συμβουλευτικής & ψυχολογικής υποστήριξης, λέσχη σίτισης, θεατρικές- χορευτικές ομάδες κλπ.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Burnette E. (2015). *Hello, Android: Introducing Google's Mobile Development Platform (Pragmatic Programmers)*. p. 7.

DiMarzio D. F. (2016). *Beginning Android Programming with Android Studio*, pp. 2-5.

Gargenta M. (2011). *Learning Android*, O'Reilly Media, United States, , pp. 31-34.

Hasenan C. (2008). *Android Essentials*, Firstpress

Lee W. M. (2011). *Beginning Android Application Development*. Indianapolis: Wiley Publishing, Incorporation.

Meier R. (2010). *Professional Android 2 Application Development*, Wrox

Morris J. (2011). *Android User Interface Development Beginner's Guide*, Pakt Publishing

Pandey G. & Dani D. (2014). *Android Mobile Application Build on Eclipse*, International Journal of Scientific and Research Publications, issue 2, 02/2014, pp. 2-3

Reto M. (2010). *Professional Android Application Development*, 1st edition, Wrox.

Singh R. (2014). *An Overview of Android Operating System and Its Security Features*, Journal of Engineering Research and Applications, issue 2, 02/ 2014, p. 519.

Silberschatz, A. & Galvin, B. P. & Gagne, G. (2005). *Operating Systems Concepts*. 7th edition. Hoboken: John Wiley & Sons, Incorporation.

Steele J. (2010). *The Android Developer's Cookbook*, Addison & Wesley

Tanenbaum, A. S. & Bos, H. (2009). *Modern Operating Systems*. 3rd edition. New Jersey: Pearson Education, Incorporation

Διαδικτυακές Πηγές

- <http://www.allaboutandroid.gr>
- <https://www.android.com>
- Android version history. [Online]. Available: https://en.wikipedia.org/wiki/Android_version_history.
- <https://www.androidpit.com>
- <https://developer.android.com/index.html>
- Droid life - Android Distribution." [Online]. Available: <http://www.droid-life.com/tag/distribution/page/2>
- Google, ART and Dalvik. [Online]. Available: <https://source.android.com/devices/tech/dalvik>

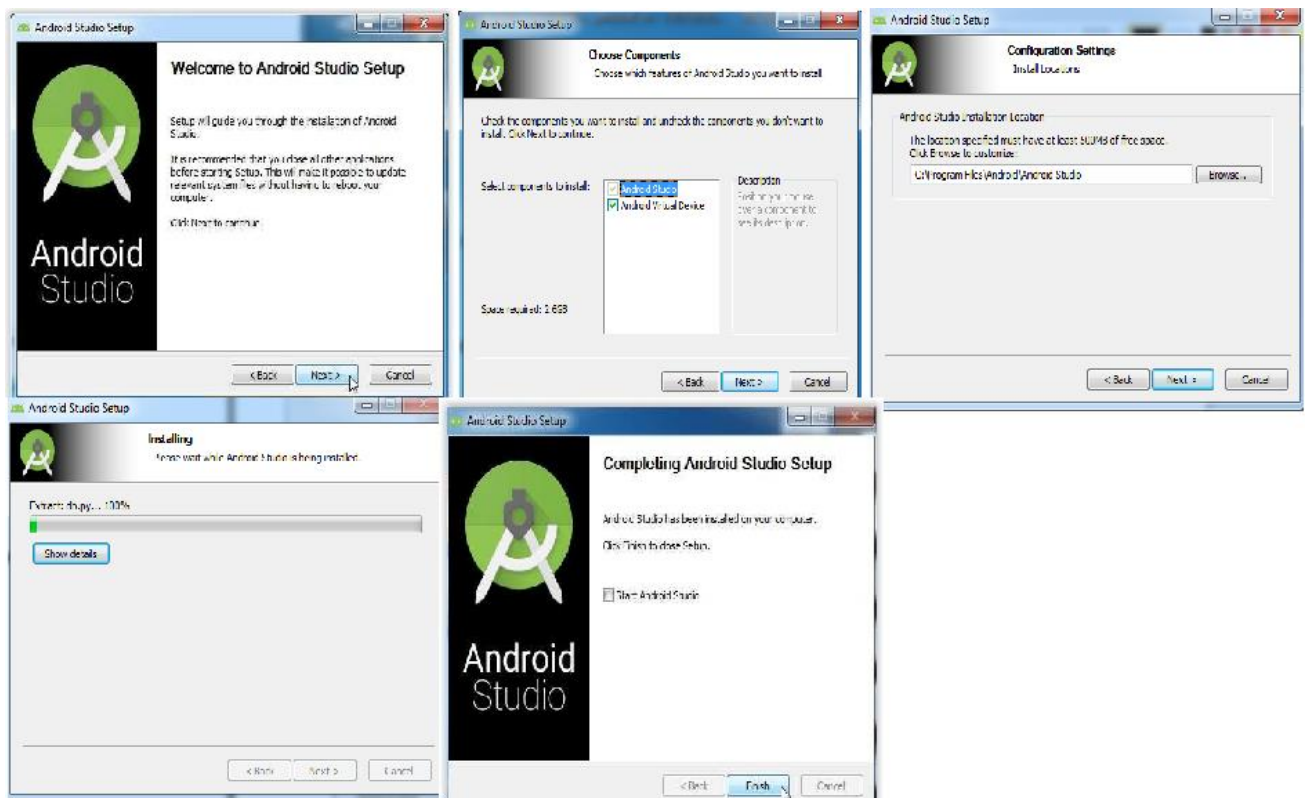
- Linuxtopia, Android Development Android Application Fundamentals . August 2012,[Online]. Available:
http://www.linuxtopia.org/online_books/android/devguide/guide/topics/fundamentals.html
- MySQL - The world's most popular open source database. [Online]. Available:
<https://www.mysql.com/>
- Netmarketshare - Operating Systems. [Online]. Available:
<https://www.netmarketshare.com>.
- Open Handset Consortium [Online]. Available:
https://www.openhandsetalliance.com/oha_overview.html.
- <https://statcounter.com>
- <http://www.vogella.com/tutorials/Android/article.html>

ΠΑΡΑΡΤΗΜΑ

1. Εγκατάσταση του Android Studio

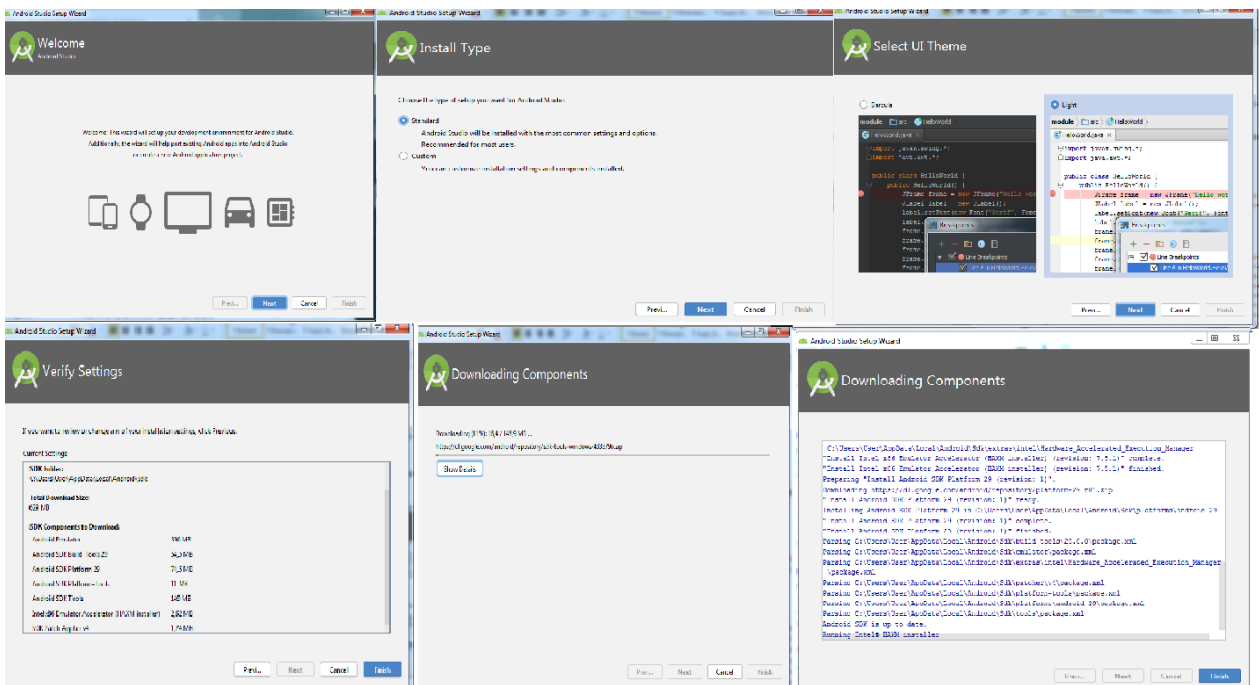
Η λήψη της εφαρμογής Android Studio γίνεται από τον ιστότοπο <https://developer.android.com/studio/>.

Εκτελούμε το αρχείο εγκατάστασης androidstudio-bundle.exe και κατά τη διάρκεια της διαδικασίας διαμορφώνουμε την εγκατάσταση ανάλογα με τις απαιτήσεις μας.

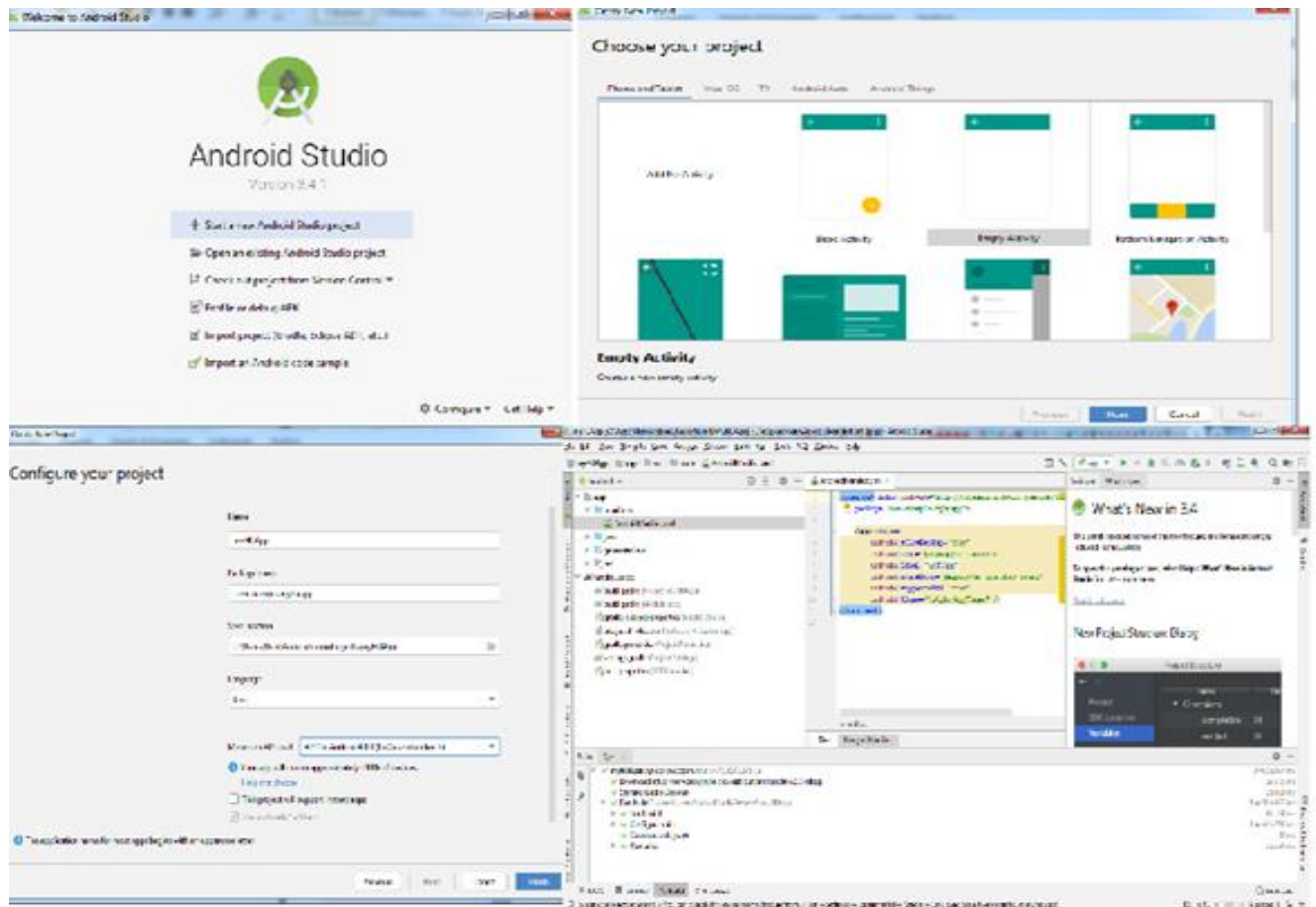


Εικόνα Π1: Στάδια εγκατάστασης android

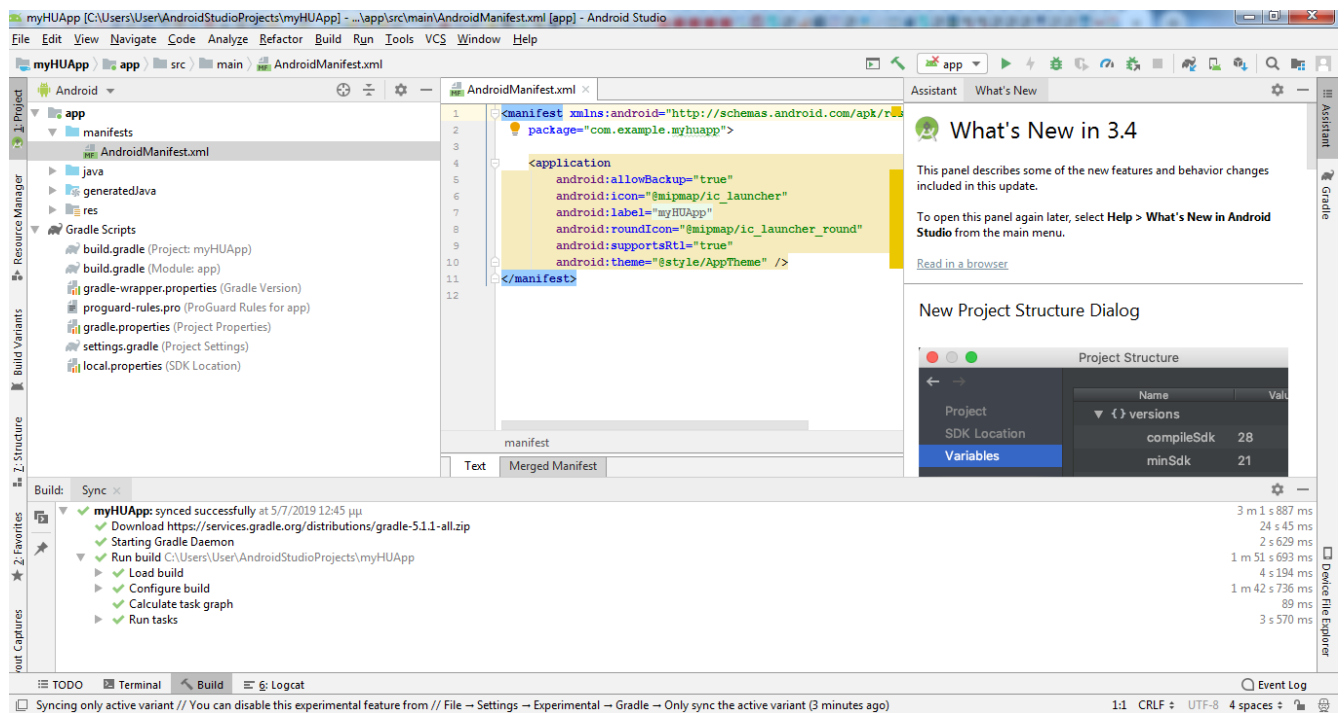
Ακολουθεί η παραμετροποίηση του λογισμικού που περιλαμβάνει τα χαρακτηριστικά διεπαφής με το χρήστη, τα πακέτα και τις βιβλιοθήκες που θα εγκατασταθούν κλπ.



Εικόνα Π2: Βήματα παραμετροποίησης διεπαφής



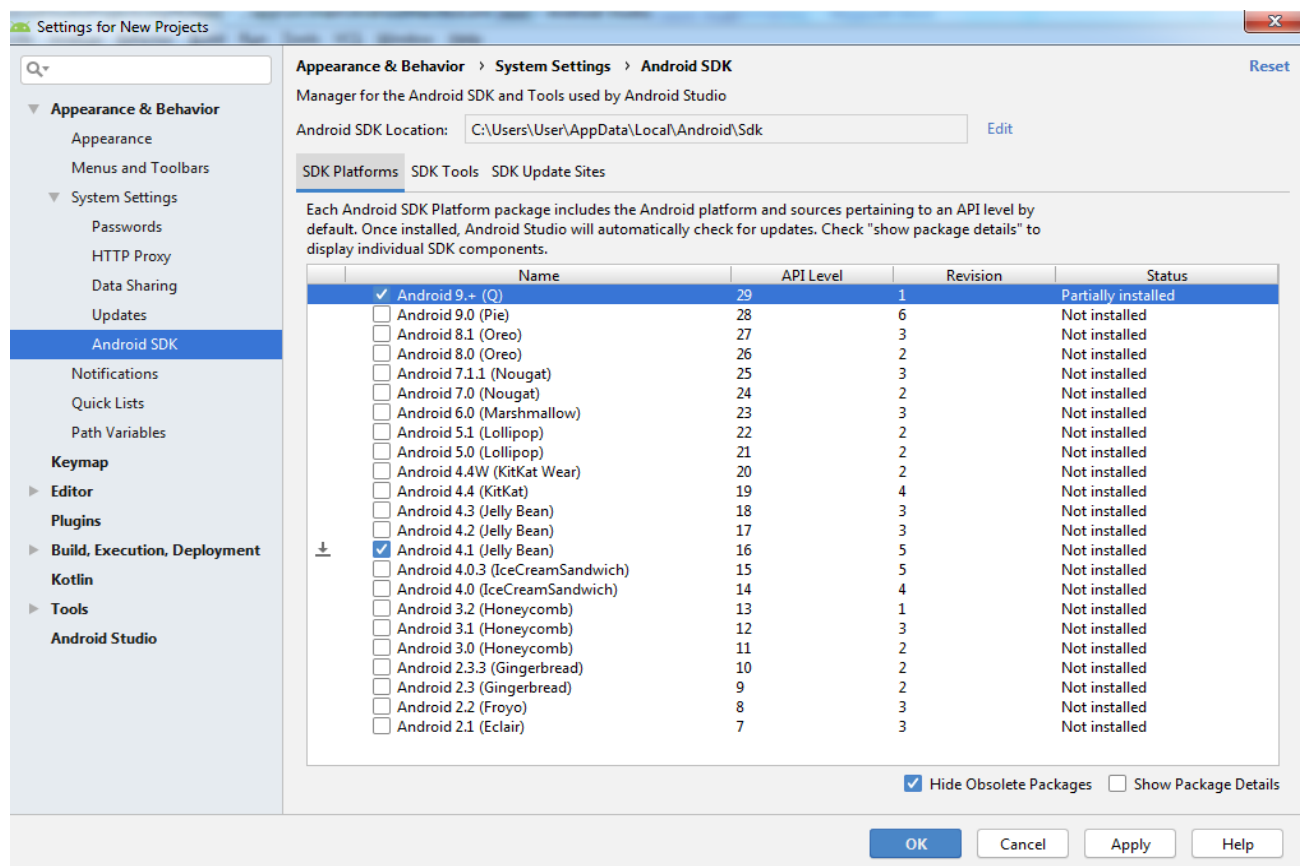
Εικόνα Π3: Διαδικασία δημιουργίας νέου σχεδίου εφαρμογής



Εικόνα Π4: Κεντρική εικόνα πλατφόρμας ανάπτυξης κώδικα

Android Standard Development Kit (SDK)

Το Android SDK αποτελεί το βασικό εργαλείο ανάπτυξης εφαρμογών και περιλαμβάνει τις βιβλιοθήκες Java, τον emulator για τη δοκιμαστική εκτέλεση, τον compiler που παράγει τον κώδικα και διάφορες βιβλιοθήκες.



Εικόνα Π5: Επιλογή πρόσθετων πακέτων SDK

Αποσφαλμάτωση (Debugging)

Το Android Studio, όπως γίνεται συνήθως με τις εφαρμογές ανάπτυξης κώδικα, παρέχει κάποια εργαλεία αποσφαλμάτωσης της εφαρμογής.

Το λειτουργικό σύστημα ορίζει ποιούς πόρους θα χρησιμοποιήσει η εφαρμογή και αυτή η διαδικασία καταγράφεται στο Logcat.

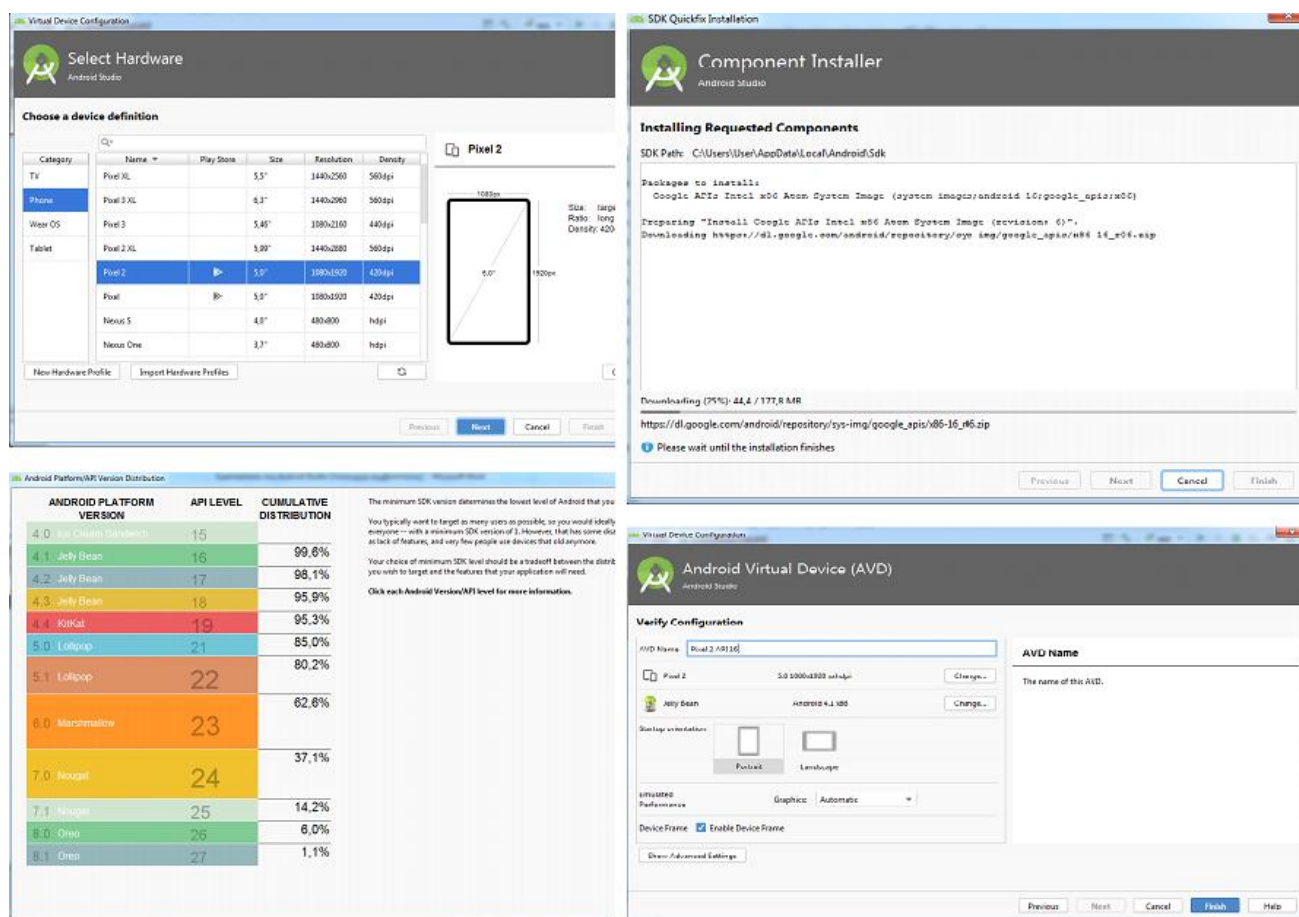
Επίσης παρέχει πληροφορίες στον προγραμματιστή για παραμέτρους όπως η μνήμη που χρησιμοποιεί η εφαρμογή, τυχόν συντακτικά και λογικά λάθη στον κώδικα, σφάλματα στο μητρώο, παραβίαση αδειών, παραβίαση δεδομένων της εφαρμογής κλπ.

Παράλληλα είναι διαθέσιμο το εργαλείο DDMS το οποίο παρέχει την δυνατότητα ελέγχου της εφαρμογής σε εικονικές συσκευές, χωρίς την ανάγκη πρόσβασης σε ζωντανά δεδομένα, για μια σειρά λειτουργιών όπως τη γεωγραφική θέση, την κλήση αριθμού, την προβολή website, την προβολή στατιστικών heap memory allocation, το thread memory size, τα network statistics και άλλα.

Android Virtual Device Manager(AVD)

Αποτελεί σημαντικό εργαλείο για τους προγραμματιστές μιας και ο AVD μας παρέχει όλα τα απαραίτητα αρχεία για την δημιουργία μιας εικονικής συσκευής με σκοπό να τεστάρουμε μια εφαρμογή.

Για να λειτουργήσει η εικονική συσκευή θα πρέπει να έχουμε κατεβάσει τα απαραίτητα αρχεία, να έχουμε εγκαταστήσει την απαιτούμενη έκδοση του API και να έχουμε επιλέξει τις κατάλληλες ρυθμίσεις συσκευής για την καλύτερη επίδοση της εφαρμογής.



Εικόνα Π6: Ρύθμιση Android Virtual Device Manager

Παράθεση τμήματος κώδικα - Υλοποίηση Κεντρικής Οθόνης

Θέλουμε η εφαρμογή μας να δίνει πρόσβαση στις υπόλοιπες δυνατότητες μέσω μερικών κουμπιών τα οποία θα είναι οργανωμένα σε πλέγμα. Επιλέχθηκε η GridView γιατί διασφαλίζει τη σωστή εμφάνιση σε όλα τα μεγέθη οθόνης.

Δημιουργούμε ένα LinearLayout και μέσα σε αυτό τοποθετούμε το GridView χρησιμοποιώντας τον παρακάτω κώδικα.

```
<GridView
```

```
android:layout_width="match_parent" android:numColumns="2" android:id="@+id/grid"
android:background="#e7e7e7" android:columnWidth="150dp"
```

```
android:gravity="center" android:verticalSpacing="10dp" android:horizontalSpacing="10dp"
android:stretchMode="columnWidth" android:layout_height="match_parent">
```

```
</GridView>
```

Το gravity δηλώνει τη τοποθέτηση του αντικειμένου του κάθε κελιού μέσα στο layout. Το numColumns δηλώνει τον αριθμό των στηλών που θα έχουμε.

Θα πρέπει όμως να δημιουργήσουμε και το xml αρχείο του GridView. Έτσι δημιουργούμε ένα νέο LinearLayout με όνομα gridAdapter και σε αυτό θα συμπεριλάβουμε ένα ImageView για την εικόνα και ένα TextView για το κείμενο. Έχουμε λοιπόν τον εξής κώδικα:

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

```
android:layout_width="350dp" android:layout_height="200dp" android:background="#ffffff"
android:elevation="5dp" android:orientation="vertical" android:padding="15dp">
```

```
<ImageView
```

```
android:id="@+id/imageHolder" android:layout_width="140dp" android:layout_height="70dp"
android:layout_gravity="center" android:layout_weight="0.21"
```

```
android:src="@drawable/ic_notes" />
```

```
<TextView
```

```

android:id="@+id/namePlacer"    android:layout_width="wrap_content"
android:layout_height="35dp" android:layout_gravity="center" android:gravity="center"

android:text="TITLE  HERE"

android:textColor="#000000" android:textSize="14dp" android:textStyle="bold" />
</LinearLayout

```

Ένα σημαντικό στοιχείο που πρέπει να λάβουμε υπόψη είναι πως προβάλλεται η GridView στους διαφορετικούς προσανατολισμούς της συσκευής (οριζόντιος- κάθετος) . Σε κάθε προσανατολισμό έχουμε διαφορετικό χώρο να χρησιμοποιήσουμε, οπότε ορίζουμε δυο ίδια base_layout με διαφορετικό προσανατολισμό, το ένα κάθετο (portrait) και το άλλο οριζόντιο (landscape).

Στην κεντρική μας activity (AppBase.class) δηλώνουμε τις τιμές που μπορεί να πάρει το TextView του GridView.

Έτσι έχουμε :

```

gridView = (GridView)findViewById(R.id.grid);  basicFields.add("ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ");
basicFields.add("ΤΜΗΜΑΤΑ"); basicFields.add("ΥΠΗΡΕΣΙΕΣ"); basicFields.add("ΕΠΙΚΟΙΝΩΝΙΑ");
basicFields.add("SOS"); basicFields.add("ΠΛΟΗΓΗΣΗ");

adapter = new gridViewAdapter(this,basicFields);

gridView.setAdapter(adapter);

```

Δημιουργούμε μια νέα class με όνομα gridViewAdapter και σε αυτή αναλογα με το τι έχουμε στο TextView τοποθετεί την αντίστοιχη εικόνα για κουμπί και καλεί την αντίστοιχη κλάση όταν πατήσουμε το κουμπί,

Έχουμε για παράδειγμα:

```

TextView textView = (TextView)v.findViewById(R.id.namePlacer);

ImageView imageView = (ImageView)v.findViewById(R.id.imageHolder);

if(names.get(position).toString().equals("ΤΜΗΜΑΤΑ"))

```

```

{

imageView.setImageResource(R.drawable.ic_tmimata);    v.setOnClickListener(new
View.OnClickListener() {

@Override

public void onClick(View v) { Intent launchinIntent = new Intent(activity, Tmimata.class);

activity.startActivity(launchinIntent);

}

```

Με τον ίδιο τρόπο λειτουργούμε σε όλες τις υπόλοιπες περιπτώσεις.

Πρόσβαση μενού "ΓΕΩΓΡΑΦΙΑΣ"

Δημιουργούμε ένα νέο Layout με όνομα geografias.xml και μέσα σε αυτό τοποθετούμε μια ListView

```

<ListView

android:id="@+id/listView"    android:layout_width="match_parent"
android:layout_height="wrap_content" />

```

Δημιουργούμε ένα layout τύπου webview με όνομα Geografia.xml

```

<WebView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/webview1"        android:layout_width="fill_parent"
android:layout_height="fill_parent"

/>

```

Δημιουργούμε μια class με όνομα Geografias.class και μέσω αυτής προβάλουμε το link του τμήματος Γεωγραφίας στο webview

```

public class Geografia extends Activity {

private WebView mWebView1;


@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.egram);

mWebView1 = (WebView) findViewById(R.id.webview1); WebSettings webSettings =
mWebView1.getSettings(); webSettings.setJavaScriptEnabled(true);


mWebView1.loadUrl("https://www.geo.hua.gr/");
mWebView1.setWebViewClient(new GeografiasClient());
}

private class GeografiasClient extends WebViewClient { @Override

public boolean shouldOverrideUrlLoading(WebView

webview1, String url) {

webview1.loadUrl(url);

return true;

}

}

public boolean onKeyDown(int keyCode, KeyEvent event)

{

if ((keyCode == KeyEvent.KEYCODE_BACK) &&

mWebView1.canGoBack())

{

```

```

mWebView1.goBack(); return true;

}

return super.onKeyDown(keyCode, event);

}

```

Ανάλογα με τι επιλέξουμε θα μας ανοίξει στον browser του κινητού το αντίστοιχο link , χρησιμοποιώντας τον εξής κώδικα:

```

protected

position,

{

void onItemClick(ListView l,

long id)

View v, int

Uri uri = Uri.parse((links.get(position)).toString());

Intent intent = new Intent(Intent.ACTION_VIEW, uri);

startActivity(intent);

}

```

Μενού "Επικοινωνίας"

Δημιουργούμε ένα LinearLayout με όνομα epikoinonia.xml και μέσα σε αυτό τοποθετούμε ένα TextView

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"

    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView

        android:layout_width="match_parent"                android:layout_height="match_parent"
        android:text="@string/epikoinonia"                android:textColor="@android:color/black"
        android:autoLink="all" android:textSize="16dp"

        android:textStyle="normal" />

    </LinearLayout>

```

Με την επιλογή `autolink="all"` κάνουμε τα διαφορα επιμέρους στοιχεία επιλέξιμα (clickable), μπορούμε έτσι να επιλέξουμε ένα νούμερο τηλεφώνου και να ανοίξει στην εφαρμογή τηλεφώνου ώστε να το καλέσουμε, επιλέγοντας ένα email να ανοίξει αυτόματα η υπηρεσία ηλεκτρονικού ταχυδρομίου που χρησιμοποιούμε ώστε να στείλουμε mail σε αυτο και τέλος ένα link όταν το πατήσουμε να μας το ανοίξει αυτόματα στον browser που έχουμε προεπιλέξει.

Έπειτα δημιουργούμε και την αντίστοιχη κλάση με όνομα `epikoinonia.class` στην οποία απλά δηλώνουμε σαν `contentview` το αντίστοιχο layout

```

public class epikoinonia extends Activity { @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.epikoinonia);
    }

}

```

Δήλωση Activities στο AndroidManifest

Όλες οι activities της εφαρμογής (καθώς και οι ιδιότητές τους) πρέπει να δηλώνονται στο αρχείο `AndroidManifest`

Αρχικά στο Android Manifest δηλώνουμε τις άδειες χρήσης:

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

Με αυτό δίνουμε άδεια στην εφαρμογή να έχει πρόσβαση στο Internet ενώ στη συνέχεια δηλώνουμε την κεντρική δραστηριότητα;

```
<activity
```

```
android:name="com.myHUApp.automation.AppBase" android:label="@string/app_name">
```

```
<intent-filter>
```

```
<action android:name="android.intent.action.MAIN" />
```

```
<category android:name="android.intent.category.LAUNCHER" />
```

```
</intent-filter>
```

```
</activity>
```

Έτσι δηλώνουμε ότι η activity AppBase είναι η κεντρική activity όταν τρέχουμε την εφαρμογή και με την παράμετρο intent-filter δηλώνουμε ότι θέλουμε να πηγαίνουμε από την μια δραστηριότητα στην άλλη μέσω Internet.

Για τις υπόλοιπες activities δηλώνουμε και ποιά είναι η parent activity αυτής. Για παράδειγμα έχουμε:

```
<activity android:name="com.myHUApp.automation.ProgrammaActivity"
android:parentActivityName="com.myHUApp.automation.AppBase" />
```

Παράθεση τμήματος Κώδικα για σύνδεση με Google Maps

Αρχικά πρέπει να λάβουμε ένα κλειδί API για να έχουμε πρόσβαση στον server του Google Map.

Στην ιστοσελίδα <https://console.developers.google.com/project> επιλέγουμε τη δημιουργία νέου project και επιλέγουμε Enable APIS AND SERICES.

Έπειτα επιλέγουμε 'Google Map Android API' και συγκεκριμένα την επιλογή 'Create Credentials' για να λάβουμε το API κλειδί.

Στο Android studio επιλέγουμε project 'Google Maps Activity' και επεξεργαζόμαστε το google_maps_api.xml (debug) (φάκελος values), για να θέσουμε το κλειδί API που δημιουργήσαμε προηγουμένως.

```
<string          name="google_maps_key"          templateMergeStrategy="preserve"
translatable="false">API_KEY</string>

</resources>
```

Ανοίγουμε το build.gradle code και προσθέτουμε το 'com.google.android.gms:play-services:8.4.0' στο dependencies

build.gradle code

```
apply plugin: 'com.android.application'
```

```
android {
    compileSdkVersion 26
    buildToolsVersion "26.0.2"
    defaultConfig {
        applicationId "com.myHUApp.GoogleMaps.googlemaps"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
```



```

        minifyEnabled false

        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
}

```

```

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:26.+'
    compile 'com.google.android.gms:play-services:8.4.0'
    testCompile 'junit:junit:4.12'
}

```

Ανοίγουμε έπειτα το activity_maps.xml και προσθέτουμε τον παρακάτω κώδικα:

activity_maps.xml code

```

<fragment android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    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"

```

```
tools:context="com.myHUApp.GoogleMaps.googlemaps.MainActivity"/>
```

Ακολουθεί ο καθορισμός αδειών internet και γεωεντοπισμού στο Android manifest:

INTERNET –ACCESS_FINE_LOCATION

AndroidManifest.xml code:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest package="com.myHUApp.GoogleMaps.googlemaps"

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

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

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

    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

        android:label="@string/app_name"

        android:supportsRtl="true"

        android:theme="@style/AppTheme">

        <meta-data

            android:name="com.google.android.geo.API_KEY"

            android:value="@string/google_maps_key"/>

        <activity

            android:name="com.myHUApp.GoogleMaps.googlemaps.MainActivity"

            android:label="@string/title_activity_maps">

            <intent-filter>

                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>

            </intent-filter>

        </activity>

    </application>

</manifest>
```

```
</intent-filter>

</activity>

</application>

</manifest>
```

Ακολουθεί προσθήκη κώδικα στο MapsActivity.java (callbacks Google Maps) :

-OnMapReadyCallback:

@Override

public void onMapReady(GoogleMap googleMap) {}

-GoogleApiClient.ConnectionCallbacks:

@Override

public void onConnected(Bundle bundle) {}

@Override

public void onConnectionSuspended(int i) {

}

-GoogleApiClient.OnConnectionFailedListener:

@Override

public void onConnectionFailed(ConnectionResult connectionResult) {

}

-LocationListener:.

@Override

public void onLocationChanged(Location location) {}

-addConnectionCallbacks():

-addOnConnectionFailedListener().

-GoogleApiClient.Builder:

-mGoogleApiClient.connect():

```

protected synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
    mGoogleApiClient.connect();
}

-Zoom Controls mMap.getUiSettings().setZoomControlsEnabled(true);

-Zoom Gestures:

Zoom In: Zoom Out:

mMap.getUiSettings().setZoomGesturesEnabled(true);

-Compass:

mMap.getUiSettings().setCompassEnabled(true);

-Changing the Map Type:

Τύποι προβολής android map:

mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);

MAP_TYPE_NORMAL :
MAP_TYPE_SATELLITE: .
MAP_TYPE_TERRAIN:
MAP_TYPE_HYBRID :
Map_TYPE_NONE :

```

@Override

```
public void onConnected(Bundle bundle) {  
    mLocationRequest = new LocationRequest();  
    mLocationRequest.setInterval(1000);  
    mLocationRequest.setFastestInterval(1000);  
    mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);  
    if (ContextCompat.checkSelfPermission(this,  
        Manifest.permission.ACCESS_FINE_LOCATION)  
        == PackageManager.PERMISSION_GRANTED) {  
        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,  
            mLocationRequest, this);  
    }  
}
```