



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή Εργασία

**“Υλοποίηση εφαρμογής σε κινητά τηλέφωνα για τις ανάγκες
του τμήματος Νοσηλευτικής του ΤΕΙ Ηπείρου”**

Μάστορας Βασίλειος

A.M: 13963



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΤΣΟΥΛΟΣ ΙΩΑΝΝΗΣ

ΑΡΤΑ ΙΑΝΟΥΑΡΙΟΣ 2018



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή Εργασία

**“Υλοποίηση εφαρμογής σε κινητά τηλέφωνα για τις ανάγκες
του τμήματος Νοσηλευτικής του ΤΕΙ Ηπείρου”**

Μάστορας Βασίλειος

A.M: 13963

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΤΣΟΥΛΟΣ ΙΩΑΝΝΗΣ

ΑΡΤΑ ΙΑΝΟΥΑΡΙΟΣ 2018



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Thesis

**“Implementation of application in mobile phones for the needs
of the Nursing Department of Technological Educational
Institute of Epirus”**

MASTORAS VASILEIOS

R.N: 13963

SUPERVISOR:

TSOULOS IOANNIS

ARTA JANUARY 2018

“Υλοποίηση εφαρμογής σε κινητά τηλέφωνα για τις ανάγκες του τμήματος Νοσηλευτικής του ΤΕΙ Ηπείρου”

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Όνομα Επίθετο,

τίτλος, βαθμίδα

2. Μέλος επιτροπής

Όνομα Επίθετο,

τίτλος, βαθμίδα

3. Μέλος επιτροπής

Όνομα Επίθετο,

τίτλος, βαθμίδα

Ο/Η Προϊστάμενος/η του

Τμήματος Όνομα Επίθετο,

τίτλος, βαθμίδα

Υπογραφή

© Επίθετο, Όνομα, έτος.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Επίθετο, Όνομα

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή, κ. Τσούλο Ιωάννη, διότι καθ' όλη την προσπάθεια μου να φέρω σε πέρας την πτυχιακή εργασία, ήταν πάντα διαθέσιμος, με τις πολύτιμες συμβουλές του αλλά και την βοήθεια του όσο περισσότερο γινόταν. Όλα αυτά, μαζί με την άψογη συνεργασία μας, συνέβαλαν στην επιτυχή ολοκλήρωσή της εργασίας. Επίσης να ευχαριστήσω όλους τους καθηγητές με τους οποίους συνεργάστηκα, κατά την διάρκεια των σπουδών μου.

Τέλος ένα μεγάλο ευχαριστώ στην οικογένεια μου, αλλά και το φιλικό μου περιβάλλον, που με την στήριξη τους σε δύσκολες στιγμές της φοίτησής μου συνολικά, κατάφερα και έφερα σε πέρας όλη αυτή την δύσκολη διαδρομή προς το πτυχίο.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας, είναι η παρουσίαση της εξέλιξης των κινητών συσκευών (κινητά τηλέφωνα) μέχρι σήμερα, που ονομάζονται smartphones και πως μπορεί κάποιος, να φτιάξει κάποια εφαρμογή για smartphones με λειτουργικό σύστημα android. Πιο συγκεκριμένα γίνεται αναφορά σε κάποιες παρόμοιες εφαρμογές με αυτή που παρουσιάζεται στην παρούσα πτυχιακή εργασία και το χρησιμοποιούμενο λογισμικό, που χρειάστηκε για την ανάπτυξη της εφαρμογής. Επίσης αναλύονται συνολικά οι κινητές συσκευές, με ιστορικά δεδομένα, την ιστορία των λογισμικών πριν την εξέλιξη των σημερινών κυρίαρχων λογισμικών και η σύγκριση μεταξύ των σημερινών λειτουργικών. Επιπλέον αναφέρεται, στο πώς μπορεί κάποιος να προγραμματίσει εφαρμογές android μέσω του λογισμικού Eclipse, την χρήση τους με το διαδίκτυο (internet) αλλά και ποιοι τρόποι αποθήκευσης δεδομένων υπάρχουν για τις εφαρμογές. Στο τέλος παρουσιάζεται η λειτουργία της εφαρμογής και οι μελλοντικές της βελτιώσεις και ο κώδικας της.

Λέξεις-κλειδιά: smartphones, διαδίκτυο, εφαρμογές.

ABSTRACT

The purpose of this dissertation is to present the evolution of the mobile devices (mobile phones) until today, which are called smartphones and the way in which someone can create an application compatible with smartphones which use an android software. In particular applications similar to the one presented in this assignment are mentioned, as well as the software that was used for the development of the application. In addition, there is an overall analysis of mobile phones along with historic data, the history of the softwares before the advancement of the dominant softwares and the comparison between the current operating systems. Moreover, the way in which someone can program android applications through the Eclipse software, their use on the internet, as well as which ways of saving data are available for the applications are being mentioned. Finally, the function of the application its future improvements and its code are being presented.

Keywords: smartphones, internet, applications.

Πίνακας περιεχομένων

1 ΕΙΣΑΓΩΓΗ.....	1
1.1 Σκοπός της εργασίας.....	1
1.2 Παρόμοια προγράμματα λογισμικού.....	1
1.3 Χρησιμοποιούμενο Λογισμικό	2
2 ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ.....	2
2.1 Οι πρώτες κινητές συσκευές	3
2.2 Ιστορική αναδρομή του Android.....	5
2.3 Διαφορές μεταξύ εκδόσεων Android.....	6
2.4 Ιστορία του iPhone.	8
2.5 Διαφορές μεταξύ εκδόσεων iPhone.	9
2.6 Windows Phone	12
2.7 Σύγκριση μεταξύ λειτουργικών συστημάτων των κινητών συσκευών.....	13
2.8 Διαδικασία ανεβάσματος.....	15
3 Ανάπτυξη Εφαρμογών σε Android.....	18
3.1 Εγκατάσταση του Android Plugin σε Eclipse.	19
3.2 Χρήση του Android στο Internet.	25
3.3 Αποθήκευση δεδομένων σε Android προγράμματα.	28
4 Η Προτεινόμενη Εφαρμογή.....	33
4.1 Η ιστοσελίδα της εφαρμογής	33
4.2 Οι χρησιμοποιούμενες web services	36
4.3 Παράδειγμα χρήσης της εφαρμογής.....	46
4.4 Μελλοντικές επεκτάσεις.....	51
5 Παράρτημα (ο κώδικας).....	55
Βιβλιογραφία	67

Πίνακας Εικόνων

Εικόνα 1 Μερίδιο αγοράς λειτουργικών συστημάτων	13
Εικόνα 2 Ασφάλεια λειτουργικών συστημάτων	14
Εικόνα 3 Απόκτηση του Eclipse IDE	19
Εικόνα 4 Πρώτη εκτέλεση του Eclipse IDE	20
Εικόνα 5 Προσθήκη και εγκατάσταση του Android Development Tools (ADT Plugin for Eclipse) ..	20
Εικόνα 6 Προσθήκη και εγκατάσταση του Android Development Tools (ADT Plugin for Eclipse) ..	21
Εικόνα 7 Εγκατάσταση του Android SDK (1)	22
Εικόνα 8 Εγκατάσταση του Android SDK (2)	22
Εικόνα 9 Εγκατάσταση εκδόσεων Android (APIs) και Extras	23
Εικόνα 10 Δημιουργία Android Virtual Device (AVD) (1)	24
Εικόνα 11 Δημιουργία Android Virtual Device (AVD) (1)	24
Εικόνα 12 Δημιουργία Android Virtual Device (AVD) (3)	25
Εικόνα 13 Json objects	29
Εικόνα 14 Json Arrays	29
Εικόνα 15 Json Values	30
Εικόνα 16 Json Strings	30
Εικόνα 17 Json Numbers	31
Εικόνα 18 Αρχική σελίδα	34
Εικόνα 19 Κατηγορίες ανακοινώσεων	34
Εικόνα 20 Υπηρεσίες τμήματος	35
Εικόνα 21 Σπουδές	35
Εικόνα 22 Ανθρώπινο δυναμικό	36
Εικόνα 23 Splash Screen	46
Εικόνα 24 Αρχική σελίδα εφαρμογής	47
Εικόνα 25 Μενού εφαρμογής	47
Εικόνα 26 Κατηγορίες ανακοινώσεων	48
Εικόνα 27 Οθόνη ανακοινώσεων	48
Εικόνα 28 Ανακοίνωση μόνο με κείμενο	49
Εικόνα 29 Ανακοίνωση με αρχεία (1)	49
Εικόνα 30 Ανακοίνωση με αρχεία (2)	50
Εικόνα 31 Χρήσιμοι Σύνδεσμοι	51
Εικόνα 32 Σπουδές	51
Εικόνα 33 Χρήσιμοι σύνδεσμοι	51
Εικόνα 34 Ανθρώπινο δυναμικό	52
Εικόνα 35 Σελίδα ανθρώπινου δυναμικού	53
Εικόνα 36 Τηλέφωνα επικοινωνίας	53
Εικόνα 37 Αποστολή email	54
Εικόνα 38 Πληροφορίες εφαρμογής	54

1 ΕΙΣΑΓΩΓΗ

1.1 Σκοπός της εργασίας

Ο σκοπός αυτής της πτυχιακής εργασίας, είναι να παρουσιάσει τις κινητές συσκευές (κινητά τηλέφωνα) όλων των κυρίαρχων λειτουργικών της αγοράς του σήμερα, δίνοντας τα μειονεκτήματα τους και τα πλεονεκτήματα τους, την σύγκριση μεταξύ τους αλλά και λειτουργικά του παρελθόντος.

Επίσης αναπτύσσεται ιδιαίτερα το λειτουργικό android, μέσω της παρουσίασης μίας εφαρμογής, που διαχειρίζεται το περιεχόμενο της ιστοσελίδας, του τμήματος Νοσηλευτικής του Τ.Ε.Ι ΗΠΕΙΡΟΥ.

Οι εφαρμογές για κινητά με λειτουργικό android δημιουργούνται σε περιβάλλοντα που υποστηρίζουν τη γλώσσα προγραμματισμού Java και ιδιαίτερα Java για android. Η ιστοσελίδα του τμήματος Νοσηλευτικής έχει δημιουργηθεί με το CMS Joomla (Content Management Systems, Σύστημα Διαχείρισης Περιεχομένου). Για να μπορεί η εφαρμογή να επικοινωνεί με την ιστοσελίδα δημιουργήθηκαν κάποια web services. Είναι αρχεία κώδικα php για τη σύνδεση της βάσης δεδομένων της ιστοσελίδας με την εφαρμογή, που τοποθετήθηκαν στον εξυπηρετητή (server) της Σ.Ε.Υ.Π Ιωαννίνων.

1.2 Παρόμοια προγράμματα λογισμικού

Υπάρχουν πάρα πολλά προγράμματα λογισμικού (ή εφαρμογές – apps) για τα κινητά τηλέφωνα, παρόμοιες σε λειτουργία με την εφαρμογή που παρουσιάζεται σε αυτή την πτυχιακή εργασία. Ο τρόπος λειτουργίας της είναι το κατέβασμα δεδομένων από την βάση δεδομένων της ιστοσελίδας και επεξεργασία και προβολή των δεδομένων. Τις βρίσκουμε στο διαδίκτυο είτε στα Play Store, App Store και Windows Store, για κινητά με λογισμικό android, ios και windows αντίστοιχα.

Τέτοιες εφαρμογές είναι το facebook, το twitter και άλλες πολλές που εξυπηρετούν χρήστες ιστοσελίδων, που προτιμάνε να τις έχουν εγκατεστημένες παρά να κάνουν περιήγηση σε αυτές μέσω του εκάστοτε φυλλομετρητή (browser) στα κινητά τους τηλέφωνα. Όλο και περισσότερες ιστοσελίδες στις μέρες μας αποκτούνε τέτοιες εφαρμογές. Κάνοντας σύνδεση λοιπόν με τα δεδομένα του λογαριασμού που χρησιμοποιεί ο χρήστης

και στην ιστοσελίδα, παρουσιάζονται τα ίδια δεδομένα για προβολή και επεξεργασία. Αυτό επιτυγχάνεται με την χρήση μίας κοινής βάσης δεδομένων ιστοσελίδας και εφαρμογής.

1.3 Χρησιμοποιούμενο Λογισμικό

Για την υλοποίηση της εφαρμογής απαιτήθηκαν τα εξής:

- Το Eclipse με το android sdk για τον προγραμματισμό της εφαρμογής για android συσκευές.
- PHP και SQL κώδικας για την υλοποίηση των web services που ενώνουν την βάση δεδομένων της ιστοσελίδας του τμήματος νοσηλευτικής του Τ.Ε.Ι ΗΠΕΙΡΟΥ με την εφαρμογή.

2 ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

Κινητό τηλέφωνο είναι το τηλέφωνο που δεν εξαρτάται από καλωδιακή σύνδεση με δίκτυο παροχής τηλεφωνίας και δεν εξαρτάται από κάποια τοπική ασύρματη συσκευή εκπομπής ραδιοφωνικού σήματος χαμηλής συχνότητας. Τα κινητά τηλέφωνα χρησιμοποιούν τεχνολογία κυψελών και εκπέμπουν σε υψηλές συχνότητες. Για την εκπομπή και λήψη των σημάτων χρησιμοποιείται αποκλειστικά ψηφιακή τεχνολογία με κωδικοποίηση. Τα κινητά όταν άρχισαν να διαδίδονται και να τα αποκτούν όλο και περισσότεροι, επικοινωνούσαν μόνο με τηλεφωνικές κλήσης και μηνύματα. Πλέον τα έξυπνα κινητά (smartphones), τα οποία είναι και το αντικείμενό αυτής της εργασίας, μπορούν να αποθηκεύσουν πολυμέσα όπως τραγούδια, φωτογραφίες, ηχογραφημένους ήχους, αλληλογραφία (email) και να επικοινωνήσουν με πολλούς τρόπους, π.χ. με την χρησιμοποίηση εφαρμογών επικοινωνίας, όπως το Messenger, το Viber και το Whatsapp απαιτώντας όμως την σύνδεση σε κάποιο δίκτυο WiFi ή δεδομένων κινητής τηλεφωνίας από κάποιο πάροχο. Σε αυτό το κεφάλαιο θα γίνει αναφορά στα πρώτα λογισμικά για έξυπνα κινητά, που δημιουργήθηκαν πριν τα android και ios. Επίσης θα γίνει ιστορική αναδρομή των σύγχρονων λειτουργικών και θα παρατεθεί η διαφορά μεταξύ των εκδόσεών τους, αλλά και η σύγκρισή τους. Τέλος, θα γίνει μια περιγραφή στον τρόπο που μπορούν να ανεβούν εφαρμογές για αυτά τα λογισμικά. ([βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[1\]](#))

2.1 Οι πρώτες κινητές συσκευές

Πριν το 1973, η κινητή τηλεφωνία περιοριζόταν σε τηλέφωνα εγκατεστημένα σε αυτοκίνητα και άλλα οχήματα. Η Motorola ήταν η πρώτη εταιρεία που παρήγαγε το πρώτο φορητό κινητό. Στις 3 Απριλίου του 1973 λοιπόν, ο Martin Cooper, ερευνητής και εκτελεστικό μέλος της Motorola, πραγματοποίησε την πρώτη κλήση μέσω κινητού τηλεφώνου. Αυτό το κινητό ζύγιζε 1,1 κιλά με 23 εκατοστά ύψος, 4,5 πλάτος και ονομαζόταν Motorola DynaTAC.

Οι γενιές των κινητών τηλεφώνων είναι τα 1G,2G,3G,4G. Τα κινητά που ανήκουν στην πρώτη γενιά (1G) είναι μέχρι το 1990.

Από εκεί και έπειτα στην δεκαετία του 1990 προέκυψαν τα συστήματα κινητής τηλεφωνίας δεύτερης γενιάς (2G). Δυο συστήματα ανταγωνίστηκαν για την υπεροχή στην παγκόσμια αγορά. Αυτά ήταν το ευρωπαϊκό πρότυπο GSM και το CDMA στις ΗΠΑ. Η διαφορά αυτών των συστημάτων με αυτά της πρώτης γενιάς, ήταν ότι χρησιμοποιούσαν ψηφιακή μετάδοση αντί για αναλογική. Το 1993 το IBM Simon ήταν ίσως το πρώτο έξυπνο κινητό. Τα κινητά δεύτερης γενιάς ήταν μικρότερα από αυτά της πρώτης. Αυτό κατέστη δυνατό μέσω τεχνολογικών βελτιώσεων, όπως πιο προηγμένων μπαταριών και ηλεκτρονικών κυκλωμάτων, με μεγαλύτερη ενεργειακή απόδοση. Η δεύτερη γενιά εισήγαγε ένα νέο τρόπο επικοινωνίας, τα γραπτά μηνύματα (SMS). Αρχικά ήταν διαθέσιμα μόνο στα δίκτυα GSM, αλλά τελικά επεκτάθηκε σε όλα τα ψηφιακά δίκτυα. Επίσης στην δεύτερη γενιά προστέθηκε η ικανότητα πρόσβασης σε πολυμέσα. Το 1998 αποθηκεύτηκε σε κινητό ήχος κλήσης.

Στις αρχές του 21^{ου} αιώνα ήρθαν τα κινητά τρίτης γενιάς (3G). Αυτή η γενιά έδινε στον χρήστη τη δυνατότητα βιντεοκλήσεων και video-streaming, μηνύματα πολυμέσων περιεχομένου MMS (Multimedia Message Service), πλοήγηση WAP και WEB με ταχύτητες που φτάνουν τα 384 kbps, υψηλές ταχύτητες ασύρματης μεταφοράς δεδομένων, υψηλής ποιότητας παιχνίδια και υπηρεσίες εύρεσης θέσεως (GPS). Μέχρι το τέλος του 2007, υπήρχαν 295 εκατομμύρια συνδρομητές σε δίκτυα 3G παγκοσμίως, γεγονός που αντιπροσώπευε το 9% της συνολικής βάσης συνδρομητών.

Από 'κει και μετά, η τέταρτη γενιά (4G) παρέχει ότι και η τρίτη, αλλά υποσχόταν δέκα φορές τις ταχύτητες της τρίτης, δηλαδή πρέπει να φτάνουν το μέγιστο 100 Mbps, για χρήση σε κινητά και για σταθερές συνδέσεις να φτάνουν το 1 Gbps. Όμως όταν ανακοινώθηκαν αυτά τα στάνταρ για πρώτη φορά, οι ταχύτητες αυτές ήταν ανήκουστες. Με

το πέρασμα του χρόνου, τα συστήματα που στηρίζουν αυτά τα δίκτυα, έφτασαν αυτές τις ταχύτητες.

Εκτός από την εξέλιξη των δικτύων, είχαμε και αυτή των λειτουργικών συστημάτων, μέχρι να φτάσουμε στα σημερινά κυρίαρχα, το ios και το android, που οι πρώτες τους εκδόσεις βγήκαν το 2007 και το 2008 αντίστοιχα.

- Το 1996 το Palm OS 1.0 με εφαρμογές για διευθύνσεις, ημερομηνίες και υπενθύμιση πληροφοριών.
- Το 1997 το Palm OS 2.0 πρόσθεσε την εφαρμογή του email.
- Το 1998 η Symbian Ltd. ανέπτυξε το λειτουργικό Symbian, το οποίο το χρησιμοποίησαν πολλές μεγάλες μάρκες τηλεφώνων, αλλά περισσότερο τα Nokia. Επίσης την ίδια χρονιά βγήκε και το Palm OS 3.0, με υποστήριξη HotSync, υποστήριξη Web Clipping και την υποστήριξη περισσότερων χρωμάτων με 8 bit.
- Το 1999 παρουσιάζεται η πλατφόρμα Nokia S40 μαζί με το Nokia 7110.
- Το 2000 το Pocket PC 2000 της Microsoft.
- Το 2001 η Microsoft βγάζει μια καινούρια έκδοση για το Pocket PC το Pocket PC 2002, προσθέτοντας την εφαρμογή MSN messenger και το Media player 8. Την ίδια χρονιά βγαίνει μια καινούρια έκδοση του Palm OS το Palm OS 4.0, προσθέτοντάς τη δυνατότητα του εξωτερικού συστήματος αρχείων και αναβαθμίζει την υποστήριξη χρωμάτων σε 8 bit.
- Το 2002 παρουσιάζονται τα πρώτα smartphones των Windows CE (Pocket PC), που κυκλοφόρησε το 2000. Την ίδια χρονιά η Palm, στην έκδοση Palm OS 5.0, προσθέτει τη λειτουργία του Bluetooth.
- Το 2003 η Microsoft βγάζει το λειτουργικό Windows Mobile 2003, με λειτουργίες Bluetooth, pocket internet explorer και windows media player 9.0.
- Το 2004 η Microsoft αναβαθμίζει το Windows Mobile 2003 σε Windows Mobile 2003 SE. Επίσης η Palm κυκλοφορεί το Palms OS cobalt, με Wi-Fi και Bluetooth συνδεσιμότητα.
- Το 2005 η εταιρία Blackberry, βγάζει το πρώτο της λειτουργικό, το Blackberry OS 4.1. Επίσης η Microsoft κυκλοφορεί το Windows Mobile 5, προσθέτοντας το windows media player 10, σύστημα τοποθεσίας (GPS),

διαχείριση της εμφάνισης του τηλεφώνου από τον χρήστη και το office mobile. ([βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[2.1, 2.2, 2.3, 2.4 2.5 2.6 2.7\]](#))

2.2 Ιστορική αναδρομή του Android

Το android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google Inc. και αργότερα από την Open Handset Alliance. Επιτρέπει στους προγραμματιστές να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών ανεπτυγμένων από τη Google. Το android είναι σχεδιασμένο κατά κύριο λόγο για συσκευές με οθόνη αφής, όπως τα smartphones και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (android tv), αυτοκίνητα (android auto) και ρολόγια χειρός (android wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί και σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, ηλεκτρονικούς υπολογιστές (π.χ. το HP slate 21) αλλά και σε άλλες ηλεκτρονικές συσκευές. Το android είναι το πιο ευρέως διαδεδομένο λειτουργικό στον κόσμο. Οι συσκευές με android έχουν περισσότερες πωλήσεις απ' όλες τις συσκευές windows phone και ios.

Η ιστορία των εκδόσεων του android ξεκίνησε με την κυκλοφορία του android beta, το Νοέμβριο του 2007. Η πρώτη εμπορική έκδοση ήταν το android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Το android είναι υπό την συνεχή ανάπτυξη της Google και της Open Handset Alliance. Από τον Απρίλιο του 2009, οι εκδόσεις του έχουν θέμα από την ζαχαροπλαστική στην κωδική ονομασία τους, εκτός από τις εκδόσεις 1.0 και 1.1 που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα. Παρακάτω εμφανίζονται αναλυτικά οι εκδόσεις, με το κωδικό τους όνομα, τον αριθμό έκδοσης τους, την ημερομηνία κυκλοφορίας και το επίπεδο API.

- Η έκδοση 1.0, κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008, με επίπεδο API 1.
- Η έκδοση 1.1, κυκλοφόρησε στις 9 Φεβρουάριου του 2009 με επίπεδο API 2.
- Η έκδοση 1.5 με όνομα cupcake, κυκλοφόρησε στις 27 Απριλίου 2009, με επίπεδο API 3.
- Η έκδοση 1.6 με όνομα donut, κυκλοφόρησε στις 15 Σεπτεμβρίου του 2009, με επίπεδο API 4.

- Οι εκδόσεις 2.0 και 2.1 με όνομα eclair, κυκλοφόρησαν στις 26 Οκτωβρίου του 2009, με επίπεδο API 5-7.
- Οι εκδόσεις 2.2 και 2.2.3 με όνομα frogo, κυκλοφόρησαν στις 20 Μαΐου του 2010, με επίπεδο API 8.
- Οι εκδόσεις 2.3 και 2.3.7 με όνομα gingerbread, κυκλοφόρησαν στις 6 Δεκεμβρίου του 2010, με επίπεδο API 9-10.
- Οι εκδόσεις 3.0 και 3.2.6 με όνομα honeycomb, κυκλοφόρησαν στις 22 Φεβρουαρίου του 2011, με επίπεδο API 11-13.
- Οι εκδόσεις 4.0 και 4.0.4 με όνομα ice cream sandwich, κυκλοφόρησαν στις 18 Οκτωβρίου του 2011, με επίπεδο API 14-15.
- Οι εκδόσεις 4.1 και 4.3.1 με όνομα jelly bean, κυκλοφόρησαν στις 9 Ιουλίου του 2012, με επίπεδο API 16-18.
- Οι εκδόσεις 4.4 και 4.4.4 με όνομα kitkat, κυκλοφόρησαν στις 31 Οκτωβρίου του 2013, με επίπεδο API 19-20.
- Οι εκδόσεις 5.0 και 5.1.1 με όνομα lollipop, κυκλοφόρησαν στις 12 Νοεμβρίου του 2014, με επίπεδο API 21-22.
- Οι εκδόσεις 6.0 και 6.0.1 με όνομα marshmallow, κυκλοφόρησαν στις 5 Οκτωβρίου του 2015, με επίπεδο API 23.
- Οι εκδόσεις 7.0 και 7.1.1 με όνομα nougat, κυκλοφόρησαν στις 22 Αυγούστου του 2016, με επίπεδο API 24.
- Η έκδοση 8 με όνομα oreo που είναι και η τρέχουσα έκδοση, κυκλοφόρησε στις 21 Αυγούστου του 2017, με επίπεδο API 26.

[\(βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[3\]\)](#)

2.3 Διαφορές μεταξύ εκδόσεων Android.

Από την έκδοση 1.5 και μετά, ξεκίνησαν να εντάσσονται στο android τα περισσότερα χαρακτηριστικά. Στην έκδοση 1.5 με το κωδικό όνομα cupcake, τα χαρακτηριστικά που εντάσσονται είναι η υποστήριξη για εικονικά πληκτρολόγια, η δυνατότητα οι χρήστες να ανεβάζουν φωτογραφίες και βίντεο στο YouTube και στο Picasa και τέλος τα widgets. Η τρέχουσα έκδοση είναι η 8.1 με κωδικό όνομα oreo, που κυκλοφόρησε στις 5 Δεκεμβρίου του 2017.

Η έκδοση 1.6 με όνομα donut άρχισε να πείθει τους περισσότερους να αφήσουν το λειτουργικό Palm και να αρχίσουν να παίρνουν στα σοβαρά το android. Αυτή η έκδοση έφερε την συμβατότητα CDMA και μια μηχανή μετατροπής κειμένου σε ομιλία.

Στις έκδοση 2.0 και 2.1 με το όνομα eclair, η φωτογραφική κάμερα υποστηρίζει πλέον φλάς, ψηφιακό ζουμ και εξισορρόπηση λευκού και δημιούργησε ένα πιο έξυπνο πληκτρολόγιο που θα μπορούσε να κάνει κάποιες προτάσεις επαφών.

Στις εκδόσεις 2.2 και 2.2.3 με όνομα frogo η Google έκανε ένα πιο γρήγορο λειτουργικό. Όμως πέρα από την ταχύτητα, ήρθε και η προσθήκη μίας γραμμής στο κάτω μέρος της οθόνης, για γρήγορη πρόσβαση στις κλήσεις και στο μενού των εφαρμογών. Συνολικά είχε αρχίσει το android να γίνεται πιο φιλικό προς το χρήστη.

Αν και στις προηγούμενες εκδόσεις είχε δοθεί βαρύτητα στη λειτουργικότητα, η Google στις εκδόσεις 2.3 και 2.3.7 με όνομα gingerbread, πήρε την απόφαση να ανασχεδιάσει το γραφικό περιβάλλον. Η διεπαφή χρήστη ανανεώθηκε με ένα πιο σκοτεινό θέμα που χάριζε πιο γρήγορες επιδόσεις του android. Επίσης προστέθηκε η υποστήριξη για το NFC, τον εξαιρετικά χρήσιμο διαχειριστή των λήψεων αλλά και βελτιωμένη αντιγραφή-επικόλληση.

Οι εκδόσεις 3.0 και 3.2.6 με όνομα honeycomb, κατά κύριο λόγο επικεντρώθηκαν σε ένα και μόνο πράγμα, τα tablets. Το ipad από την Apple κυκλοφόρησε λίγους μήνες αργότερα και το android χρειαζόταν ένα λειτουργικό που θα μπορούσε να το ανταγωνιστεί. Αυτή η έκδοση περιελάμβανε ολογραφική διεπαφή και ένα πιο έξυπνο πληκτρολόγιο για μεγαλύτερες συσκευές.

Στις εκδόσεις 4.0 και 4.0.4 με όνομα ice cream sandwich, η γραφική διεπαφή αναπροσανατολίστηκε προς τα δεδομένα κινητού τηλεφώνου και την πλήρη ανεξάρτηση από τα υλικά κουμπιά (hardware buttons). Αυτό σημαίνει ότι δημιουργήθηκε η γραμμή ενεργειών (action bar) και σε αυτήν προστέθηκε το κουμπί εμφάνισης πρόσφατων εφαρμογών. Ο ανασχεδιασμός αυτών των εκδόσεων, ήταν μία συγχώνευση των εκδόσεων honeycomb, gingerbread και frogo κ.τ.λ. .

Στις εκδόσεις 4.1 και 4.3.1 με όνομα jelly bean, ο βοηθός φωνής βελτιώθηκε. Η αρχική οθόνη αναβαθμίστηκε, καθώς επίσης και τα widgets μπορούσαν να αλλάξουν μέγεθος και να τοποθετηθούν οπουδήποτε και γενικά το android ήταν πολύ προσαρμόσιμο.

Στις εκδόσεις 4.4 και 4.4.4 με όνομα kitkat, μειώθηκαν οι απαιτήσεις συστήματος και ακόμα συσκευές με μνήμα RAM στα 512 MB, θα μπορούσαν να λειτουργήσουν ομαλά. Από εκείνο το σημείο και μετά ξεκίνησε και το πρόγραμμα εκδόσεων Android One για πιο φθηνές συσκευές με περιορισμένους πόρους.

Με τις εκδόσεις 5.0 και 5.1.1 με όνομα lollipop, ξεκίνησε το σύγχρονο android. Το λειτουργικό επανασχεδιάστηκε πλήρως, αλλάζοντας τα πάντα. Στην οθόνη κλειδώματος ενσωματώθηκαν οι ειδοποιήσεις, κάνοντας την πιο χρήσιμη. Επίσης προστέθηκε η λειτουργία για την σίγαση των ειδοποιήσεων.

Στις εκδόσεις 6.0 και 6.0.1 με όνομα marshmallow, δεν έγιναν μεγάλες σχεδιαστικές αλλαγές, παρόλα αυτά έγιναν κάποιες εκπληκτικές προσθήκες, όπως ενός νέου τρόπου διαχείρισης των δικαιωμάτων των εφαρμογών και υποστήριξη δακτυλικών αποτυπωμάτων. Επίσης βελτιώθηκε η διάρκεια ζωής της μπαταρίας.

Στις εκδόσεις 7.0 και 7.1.1 με όνομα nougat, προστέθηκε υποστήριξη πολλαπλών παραθύρων, δηλαδή η χρήση εφαρμογών ταυτόχρονα σε διαχωρισμένη οθόνη. Το κέντρο ειδοποιήσεων ανασχεδιάστηκε .

Στην τελευταία έκδοση 8 με όνομα oreo, η ταχύτητα του λειτουργικού αυξήθηκε. Ένα σημαντικό χαρακτηριστικό είναι ότι περιορίζει τη δραστηριότητα των εφαρμογών στο παρασκήνιο. Μια άλλη καινούρια προσθήκη είναι η λειτουργία Picture In Picture. Έτσι ο χρήστης μπορεί να χρησιμοποιήσει δύο εφαρμογές, την μία μέσα στην άλλη. Επίσης με την άδεια του χρήστη, γίνεται αυτόματη συμπλήρωση των στοιχείων λογαριασμών σε εφαρμογές. Τώρα όσον αφορά τις εφαρμογές, έγιναν κάποιες αλλαγές, όπως η δυνατότητα του χρήστη να μεταβαίνει σε νέες εφαρμογές, κατευθείαν από τον browser, χωρίς να χρειάζονται εγκατάσταση, τα λεγόμενα Instant Apps. Όσον αφορά με τις εγκατεστημένες εφαρμογές, μπήκε η λειτουργία notification dots, που εμφανίζει έναν μικρό κύκλο στην πάνω δεξιά γωνία του εικονιδίου της εφαρμογής, που με το πάτημά της εμφανίζει ειδοποιήσεις, π.χ. στην εφαρμογή του Messenger κάποια μηνύματα. Τέλος το android oreo βελτίωσε τη διάρκεια ζωής της μπαταρίας. ([βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[4.1, 4.2\]](#))

2.4 Ιστορία του iPhone.

Το iPhone συγκαταλέγεται στην κατηγορία των έξυπνων κινητών (smartphones). Δημιουργήθηκε από την εταιρεία Apple, η οποία το κυκλοφόρησε στην αγορά των ΗΠΑ στις 29 Ιουνίου 2007. Έχει κυκλοφορήσει στις περισσότερες χώρες της Ευρώπης και στην Ιαπωνία.

Η πρώτη έκδοση του διέθετε τετραπλή ζώνη (quad band) με τεχνολογία EDGE. Οι χώρες που το διέθεσαν στην αγορά μετά τις ΗΠΑ ήταν το Ηνωμένο Βασίλειο, η Γαλλία και η Γερμανία, το Νοέμβριο του 2007 ενώ η Ιρλανδία και η Αυστρία το διέθεσαν την άνοιξη του επομένου έτους. Έναν χρόνο μετά την κυκλοφορία του, τον Ιούλιο του 2008,

κυκλοφόρησε το iPhone δεύτερης γενιάς με όνομα iPhone 3G, που περιελάμβανε την υποστήριξη κινητής τηλεφωνίας 3G, σε 22 χώρες, ενώ αργότερα ξεπέρασαν τις 80. Ακόμα μια εξέλιξη του iPhone ήταν το iPhone 3GS, που κυκλοφόρησε τον Ιούνιο του 2009 σε Αμερική και Ευρώπη. Στις 24 Ιουλίου του 2010 ο CEO της Apple ανακοίνωσε την τέταρτη γενιά του iPhone με ονομασία iPhone 4. Έπειτα μετά από έναν χρόνο, τον Φεβρουάριο, βγήκε στην κυκλοφορία το iPhone 4S, το τελευταίο που ξεκίνησε ο CEO της Apple, Steve Jobs, αφού στις 5 Οκτωβρίου πέθανε από καρκίνο. Στις 12 Σεπτεμβρίου του 2012 μία νέα έκδοση το iPhone 5 βγήκε στην αγορά, έπειτα το 5S στις 10 Σεπτεμβρίου του 2013 και το 5C 20 Σεπτεμβρίου του 2013. Το 2014 στις 9 Σεπτεμβρίου η έκδοση με όνομα iPhone 6 μαζί με το 6 Plus ανακοινώθηκαν. Μετά από ένα χρόνο ανακοινώθηκαν τα iPhone 6S και 6S Plus. Το 2016 στις 21 Μαρτίου, βγήκε στην αγορά μια ενδιάμεση έκδοση, το SE, που είχε τον σχεδιασμό του 5S και τα χαρακτηριστικά του 6S, ενώ στις 7 Σεπτεμβρίου του ίδιου χρόνου κυκλοφόρησαν τα μοντέλα 7 και 7 Plus. Τέλος στις 12 Σεπτεμβρίου του 2017 κυκλοφόρησαν τα iPhone 8, 8 Plus και X. ([βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[5\]](#))

2.5 Διαφορές μεταξύ εκδόσεων iPhone.

Το iOS είναι ένα λειτουργικό σύστημα για κινητές συσκευές, που δημιουργήθηκε από την Apple Inc., για τις συσκευές της iPhone, iPad και iPod Touch. Οι εκδόσεις για το iOS, διατίθενται μέσω του iTunes και από το iOS 5 και μετά, μέσω ενημερώσεων λογισμικού. Η τρέχουσα έκδοση είναι το iOS 11.2.2 που κυκλοφόρησε στις 8 Ιανουαρίου του 2018. Ακολουθούν όλες οι εκδόσεις του iOS και οι διαφορές τους,

- Η έκδοση iPhone OS 1, κυκλοφόρησε στις 29 Ιουνίου του 2007, με ορισμένα χαρακτηριστικά της την προσθήκη ατόμων στα αγαπημένα και την ομαδική συνομιλία.
- Η έκδοση iPhone OS 2, κυκλοφόρησε στις 11 Ιουλίου του 2008. Σε αυτή την έκδοση εισάγεται το AppStore.
- Η έκδοση iPhone OS 3, κυκλοφόρησε στις 17 Ιουνίου του 2009. Αυτή η έκδοση πρόσθεσε χαρακτηριστικά όπως, αντιγραφή και επικόλληση και τα MMS.
- Η έκδοση iOS 4, κυκλοφόρησε στις 21 Ιουνίου του 2010. Σε αυτή την έκδοση μπήκαν τα χαρακτηριστικά multitasking και άμεση βιντεοκλήση.

- Η έκδοση iOS 5, κυκλοφόρησε στις 6 Ιουνίου του 2011. Σε αυτή την έκδοση μπήκαν τα χαρακτηριστικά όπως κέντρο ειδοποιήσεων και η δυνατότητα αποστολής μηνυμάτων μέσω Wi-Fi για όλες τις συσκευές που έτρεχαν iOS 5. Επίσης προστέθηκε η λειτουργία υπενθυμίσεων.
- Η έκδοση iOS 6, κυκλοφόρησε στις 11 Ιουνίου του 2012. Σε αυτή την έκδοση μπήκε το χαρακτηριστικό της Siri, του ψηφιακού βοηθού. Οι χάρτες της Google αντικαταστάθηκαν από τα Apple Maps. Ένα ενδιαφέρον χαρακτηριστικό που έφερε αυτή η έκδοση ήταν η λειτουργία «μην ενοχλείτε», επιτρέποντας στους χρήστες να αποφεύγουν τις κλήσεις και τις ειδοποιήσεις, εκτός κι αν οι καλούντες είναι στην ομάδα «αγαπημένα».
- Η έκδοση iOS 7, κυκλοφόρησε στις 10 Ιουνίου του 2013. Σε αυτή την έκδοση η διεπαφή χρήστη ανασχεδιάστηκε πλήρως. Ένα ενδιαφέρον καινούριο χαρακτηριστικό είναι το AirDrop, που επιτρέπει στους χρήστες να μοιραστούν στη στιγμή φωτογραφίες, βίντεο, έγγραφα κ.τ.λ. , με άλλες συσκευές της Apple σε κοντινή απόσταση. Επίσης μέσω AppStore μπορούν οι εφαρμογές να αναβαθμίζονται και αυτόματα.
- Η έκδοση iOS 8, κυκλοφόρησε στις 2 Ιουνίου του 2014. Σε αυτή την έκδοση ένα νέο χαρακτηριστικό έκανε την εμφάνιση του, το Continuity, που επέτρεπε την επικοινωνία μεταξύ συσκευών σε διαφορετικές κατηγορίες προϊόντων της Apple δηλαδή Mac ή iPad, π.χ. αν το iPhone με iOS 8 δέχεται κλήση ή κάποιο μήνυμα και δεν το έχει δίπλα του ο χρήστης, αλλά έχει ένα Mac ή iPad, μπορεί να απαντήσει από αυτά, εφόσον βρίσκονται στο ίδιο δίκτυο Wi-Fi. Ένα άλλο καινούριο χαρακτηριστικό ήταν το Handoff, που έδινε τη δυνατότητα π.χ. ο χρήστης να γράφει ένα email στο iPhone με iOS 9 και να το συνεχίσει από εκεί που το είχε αφήσει σε ένα Mac, ή ακόμα να είναι σε μία ιστοσελίδα στο Mac του και να συνεχίσει τη περιήγηση του στο iPad, αρκεί όλες οι Apple συσκευές να είναι συγχρονισμένες στον ίδιο λογαριασμό iCloud.
- Η έκδοση iOS 9, κυκλοφόρησε στις 8 Ιουνίου του 2015. Σε αυτή την έκδοση μπήκαν αρκετά νέα χαρακτηριστικά. Αρχικά το proactive

assistance, που συνδυάζει την ψηφιακή βοηθό Siri με τη λειτουργία αναζήτησης spotlight, που ψάχνει την συσκευή για τραγούδια, επαφές κ.τ.λ. . Επίσης η έκδοση αυτή έφερε μία νέα προεγκατεστημένη εφαρμογή ειδήσεων, με όνομα News. Αυτή η εφαρμογή εμφανίζει άρθρα από πολλές πηγές στο διαδίκτυο, με βάση τα ενδιαφέροντα του κάθε χρήστη. Η έκδοση iOS 9 αναβάθμισε και την εφαρμογή των σημειώσεων, προσθέτοντας δυνατότητες όπως, δημιουργία σημειώσεων με check boxes, ενσωμάτωση φωτογραφιών και συνδέσμων URL, λήψεις δεδομένων από άλλες εφαρμογές και συγχρονισμός των σημειώσεων μέσω του iCloud σε συσκευές iPad και Mac.

- Η έκδοση iOS 10, κυκλοφόρησε στις 13 Ιουνίου του 2016. Ένα από τα καινούρια χαρακτηριστικά της νέας έκδοσης είναι, η αντικατάσταση της δυνατότητας «Slide to unlock (σύρετε για να ξεκλειδώσετε)», από την οθόνη κλειδώματος με την λειτουργία Touch Id. Επίσης η οθόνη κλειδώματος επανασχεδιάστηκε και οι ειδοποιήσεις σε αυτή έγιναν αναλυτικότερες, επιτρέποντας στον χρήστη να αλληλεπιδρά με αυτές, χωρίς να ξεκλειδώσει το κινητό. Μια από τις πιο σημαντικές αλλαγές είναι αυτή της κατάργησης εγκατάστασης, των προεγκατεστημένων εφαρμογών της Apple, κάτι που οι χρήστες το ζητούσαν επίμονα. Ακόμα προστέθηκε μια ακόμα δυνατότητα στην ψηφιακή βοηθό Siri, η οποία πλέον μπορεί να μετατρέψει οποιαδήποτε φωνητικό μήνυμα σε κείμενο. Ακόμη μια σημαντική προσθήκη είναι η λήψη κλήσεων από την οθόνη κλειδώματος, από εφαρμογές όπως το skype, viber κ.τ.λ. , χωρίς να χρειάζεται να ανοίξει η εκάστοτε εφαρμογή.
- Η έκδοση iOS 11, κυκλοφόρησε στις 5 Ιουνίου του 2017, που είναι και το τρέχον λειτουργικό για τα iPhone και iPad. Σε αυτή την έκδοση έγιναν αρκετές αλλαγές στο iPhone και ακόμα περισσότερες στο iPad. Το iOS 11 πραγματοποιεί σημαντικές αλλαγές στο λογισμικό. Η οθόνη κλειδώματος και το κέντρο ειδοποιήσεων συνδυάζονται, επιτρέποντας σε όλες τις ειδοποιήσεις να εμφανίζονται απευθείας στην οθόνη κλειδώματος, με νέες δυνατότητες για κύλιση προς τα πάνω και προς τα κάτω, για εμφάνιση και απόκρυψη περιεχομένου. Οι διαφορετικές καρτέλες στο Κέντρο Ελέγχου, συνδυάζονται σε μία, ενώ υπάρχει και

η δυνατότητα εμφάνισης περισσότερων επιλογών, με τη χρήση 3D Touch στα εικονίδια (με παρατεταμένο πάτημα μίας επιλογής εμφανίζεται ένα μενού επιλογών), καθώς και η δυνατότητα εξατομίκευσης του Κέντρου Ελέγχου. Το App Store ανανεώθηκε πλήρως. Προστέθηκε μία καινούρια εφαρμογή με όνομα «Αρχεία», που συγκεντρώνει όλα τα αρχεία του χρήστη σε ένα σημείο, καθώς και αρχεία που βρίσκονται σε iPad και σε άλλες συσκευές iOS, αν διαθέτει ο χρήστης, και σε εφαρμογές όπως iCloud drive ή DropBox κ.τ.λ. . Όπως αναφέρθηκε παραπάνω, η νέα έκδοση φέρνει μεγάλες αλλαγές και στο iPad, όπως το καινούριο Dock στο κάτω μέρος της οθόνης, που είναι διαθέσιμο σε οποιαδήποτε οθόνη, δίνοντας τη δυνατότητα εναλλαγής εφαρμογών εύκολα και γρήγορα. Το νέο αυτό Dock εξατομικεύεται κατά την επιθυμία του χρήστη, με βάση αγαπημένες εφαρμογές, πρόσφατες εφαρμογές που ανοίξαν ή που είναι αυτή την στιγμή ανοιχτές στο iPhone ή το Mac. Τέλος η Siri, η ψηφιακή βοηθός, μπορεί να μεταφράσει μεταξύ διαφορετικών γλωσσών. [\(βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[6.1\], \[6.2\], \[6.3\], \[6.4\], \[6.5\], \[6.6\], \[6.7\], \[6.8\], \[6.9\], \[6.10\], \[6.11\], \[6.12\]\)](#)

2.6 Windows Phone

Τα Windows Phone είναι η έκδοση του λειτουργικού συστήματος της Microsoft για έξυπνα κινητά. Αναπτύχθηκε στις αρχές του 2000, ενώ στις πρώτες εκδόσεις του είχε το όνομα Windows Mobile. Οι πιο πολλές συσκευές με λειτουργικό Windows είναι αυτές της Nokia, με όλη την πρώην σειρά μοντέλων Lumia. Η Microsoft εξαγόρασε τη Nokia. Πλέον η συνεργασία τους έχει σταματήσει, με τη Nokia να στρέφεται στην παραγωγή smartphones με λειτουργικό σύστημα Android και η Microsoft κράτησε την ονομασία Lumia δημιουργώντας δικές της συσκευές.

Ακολουθεί μία σύντομη ιστορική αναδρομή των εκδόσεων του λειτουργικού συστήματος Windows Phone.

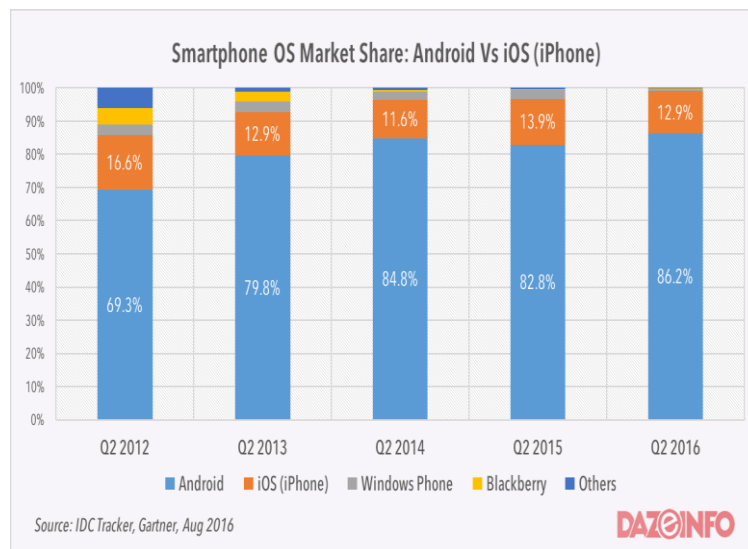
- Το Windows Phone 7, κυκλοφόρησε στις 8 Νοεμβρίου 2010 στις Ηνωμένες Πολιτείες.

- Το Windows Phone 7.5, κυκλοφόρησε το 2011.
- Το Windows Phone 7.8, κυκλοφόρησε στις 29 Οκτωβρίου 2012 και είναι η αναβαθμισμένη έκδοση του 7.5.
- Το Windows Phone 8, κυκλοφόρησε το 2013 και είχε σαν κυριότερη αλλαγή την αύξηση των εφαρμογών.
- Το Windows Phone 8.1, κυκλοφόρησε στις 10 Απριλίου του 2014.
- Το Windows Phone 10, κυκλοφόρησε στις 21 Ιανουαρίου 2015 και η κύρια αλλαγή του ήταν η ενοποίηση του με τα Windows 10 για υπολογιστές.

(βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ [7])

2.7 Σύγκριση μεταξύ λειτουργικών συστημάτων των κινητών συσκευών

Τα τελευταία χρόνια, το τμήμα των smartphones και οι εταιρίες που παράγουν λειτουργικά συστήματα για αυτά, βρίσκονται σε συνεχή ανταγωνισμό. Οι κυρίαρχοι του τομέα αναμφίβολα είναι τα λειτουργικά συστήματα Android και iOS και ποιο πίσω τα Windows phone κ.α. , με τους χρήστες ανά τον κόσμο να προτιμούν ένα από τα δύο. Τα Android και iOS αντιπροσωπεύουν περισσότερο από το 99% της αγοράς στις πωλήσεις smartphones. Όλα τα λειτουργικά έχουν τα υπέρ και τα κατά τους, με τους χρήστες να



Εικόνα 1 Μερίδιο αγοράς λειτουργικών συστημάτων

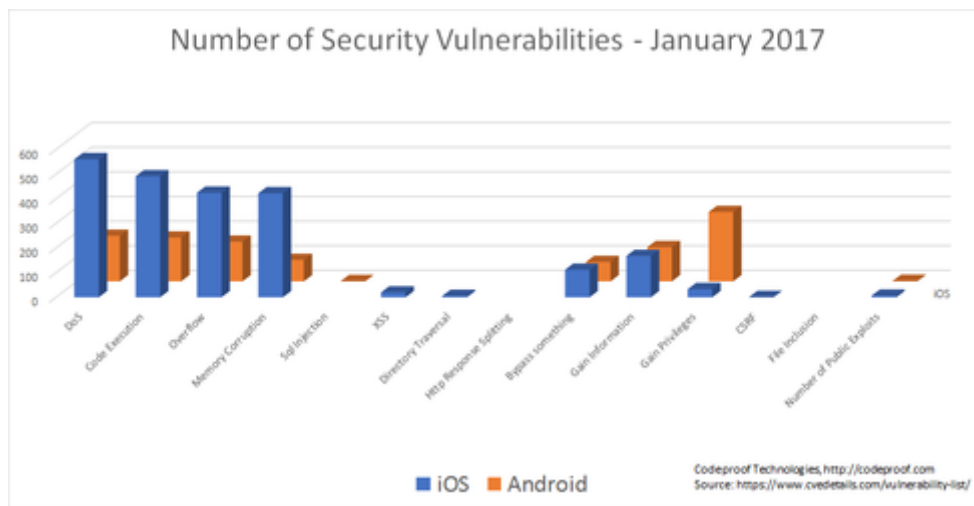
προσπαθούν να τα αντισταθμίζουν, έτσι ώστε να πάρουν την απόφασή τους στο τι τους ταιριάζει καλύτερα. Στη συνέχεια την ενότητας αυτής, αναφέρονται τα ποσοστά αγοράς για το κάθε λογισμικό με την βοήθεια γραφήματος, και μία σύντομη σύγκριση μεταξύ τους.

Όπως βλέπουμε από την έρευνα που διεξήχθη τον

Αύγουστο του 2016 (Εικόνα 1), το μεγαλύτερο ποσοστό της αγοράς καταλαμβάνει το Android με το ποσοστό του το 2016 να φτάνει στο 86,2%. Το iOS την ίδια περίοδο είχε ποσοστό 12,9%, πολύ μικρότερο από αυτό του Android, κάτι που σίγουρα δεν οφείλεται

στην ανεπάρκειά του σαν λειτουργικό, αλλά στο ότι το Android είναι λειτουργικό σύστημα για τις περισσότερες μάρκες smartphones. Σε αντίθεση το iOS είναι λειτουργικό μόνο για τις συσκευές αυτές που παράγει η ίδια η Apple. Τελευταίο έρχεται στην βασική σύγκριση αυτής της εργασίας το Windows Phone, με κάτι λιγότερο από 1%. Στο ίδιο διάγραμμα φαίνεται και η προτίμηση των χρηστών από το 2012 μέχρι το 2016. Το Android έχει μεγάλη αύξηση, έναντι του iOS και κάποιον άλλων λειτουργικών, όπως το Windows Phone που μπήκε στο «παιχνίδι» το 2010.

Το πρώτο κριτήριο ως προς την σύγκρισή στα δύο βασικά λειτουργικά, είναι η ασφάλεια δεδομένων, συμπέρασμα το οποίο βγαίνει από το παραπάνω διάγραμμα (Εικόνα 2). Εδώ έχουμε ξεκάθαρο νικητή το iOS, όπως φαίνεται, και ο λόγος είναι ότι, είναι λογισμικό κλειστού κώδικα, ενώ το Android είναι ανοιχτού, κάτι το οποίο το καθιστά ελαφρώς πιο ευάλωτο σε απειλές.



Εικόνα 2 Ασφάλεια λειτουργικών συστημάτων

Ένα άλλο κριτήριο είναι η διαθεσιμότητα των εφαρμογών που προσφέρουν το Play Store, App Store και Windows Store. Μεταξύ του Play Store και του App Store, των Android και iOS αντίστοιχα, και τα δύο διαθέτουν πληθώρα εφαρμογών. Οι περισσότερες εφαρμογές είναι χωρίς πληρωμή στο Play Store, εν αντιθέσει με το App Store. Τέλος το Windows Store στο μόνο που κερδίζει, είναι το Microsoft Office, επειδή το Windows Phone είναι λειτουργικό της Microsoft. Κατά τα άλλα οι συσκευές αυτές δεν είναι ιδιαίτερα εύχρηστες, αν και μετά από την έκδοση 8.1 που μοιάζουν πολύ με τις αντίστοιχες εκδόσεις για τους H/Y, κάπως έγιναν πιο φιλικές προς τον χρήστη. Το μεγαλύτερο τους όμως μειονέκτημα είναι ότι το Windows Store, διαθέτει πολύ λίγες εφαρμογές και πολλές από αυτές είναι beta (δοκιμαστικές), όπως το Facebook App και το Messenger. (βλ. [ΒΙΒΛΙΟΓΡΑΦΙΑ \[8\]](#))

2.8 Διαδικασία ανεβάσματος

Υπάρχουν συγκεκριμένα βήματα που πρέπει να γίνουν, ούτως ώστε να ανεβεί μία εφαρμογή που δημιούργησε ένας προγραμματιστής, στο Play Store, App Store και Windows Store. Παρακάτω αναφέρονται τα βήματα και για τις τρεις περιπτώσεις.

Play Store:

1. Αρχικά πρέπει να γίνει δημιουργία λογαριασμού Developer. Η εγγραφή γίνεται εδώ (<https://play.google.com/apps/publish/signup/>), γίνεται η συμπλήρωση των στοιχείων, μετά η αποδοχή των όρων και μετά πρέπει να πληρωθεί το αντίτιμο των 20 ευρώ, μέσω του Google Wallet.
2. Στην συνέχεια επιλέγουμε «Add New Application» και στη συνέχεια γίνεται η επιλογή της γλώσσας της εφαρμογής, όπως και το όνομά της και μετά πατάμε «Prepare Store Listing».
3. Μετά πρέπει να δημιουργηθεί η περιγραφή της εφαρμογής, που πρέπει να είναι αναλυτική και να παραθέτει τις λειτουργίες της. Το «Promo Text» είναι ένα σύντομο κείμενο διαφήμισης της εφαρμογής. Στο «Recent Changes» γράφουμε τις αλλαγές που θα κάνουμε στην εφαρμογή σε μελλοντικά Updates. Στο Promo Text υπάρχει η δυνατότητα μέσω της επιλογής «Add Translations», να βάλουμε τα παραπάνω και σε άλλες γλώσσες.
4. Χρειάζονται τουλάχιστον 2 Screenshots από την εφαρμογή, αλλά και το εικονίδιο της εφαρμογής. Τέλος υπάρχει και η δυνατότητα να ανεβεί κάποιο βίντεο, με την χρήση της εφαρμογής YouTube.
5. Μετά γίνεται η επιλογή, αν πρόκειται για εφαρμογή ή παιχνίδι, αλλά και η κατηγορία π.χ. διασκέδαση, καθώς και για το κοινό που είναι κατάλληλη.
6. Ήρθε η ώρα να ανεβεί η εφαρμογή. Επιλέγουμε από την αριστερή στήλη το «APK» και στη συνέχεια ανεβάζουμε την εφαρμογή. Έπειτα βάση των παραμέτρων που έχουμε ορίσει στο AndroidManifest.xml, εμφανίζεται ο αριθμός διαφορετικών συσκευών που υποστηρίζει την εφαρμογή. Υπάρχει η δυνατότητα να εξαιρεθούν μία ή περισσότερες συσκευές.
7. Τέλος επιλέγουμε τον τρόπο διάθεσης της εφαρμογής. Πηγαίνουμε στα αριστερά και διαλέγουμε «Pricing And Distribution». Εκεί διαλέγουμε αν η

εφαρμογή θα είναι δωρεάν ή επί πληρωμή (από την Ελλάδα μόνο δωρεάν επιτρέπεται), καθώς και στις χώρες που θέλουμε να είναι διαθέσιμη.

App Store:

Το κόστος για να ανεβεί μία εφαρμογή στο App Store είναι 99 ευρώ.

1. Συγκεντρώνουμε τις πληροφορίες της εφαρμογής, όπως screenshots, όνομα, περιγραφή, τις λέξεις κλειδιά, το εικονίδιο της εφαρμογής, την κατηγορία, τη βαθμολογία μέσω βάση ενός ερωτηματολογίου και τα url υποστήριξης και μάρκετινγκ.
2. Δημιουργία αναγνωριστικού πακέτου (Bundle Identifier), πηγαίνοντας στην ιστοσελίδα (<https://developer.apple.com/account/overview.action>) και κάνουμε σύνδεση. Πατάμε «Identifiers» και μετά το «+» στην πάνω δεξιά γωνία της οθόνης και ονομάζουμε το αναγνωριστικό εφαρμογής (Bundle ID) δίνοντας το ίδιο όνομα με την εφαρμογή. Μετά επιλέγουμε τις υπηρεσίες που χρειάζεται η εφαρμογή και πατάμε «υποβολή».
3. Στη συνέχεια δημιουργούμε αίτημα υπογραφής πιστοποιητικού (Certificate Signing Request). Ανοίγουμε το πρόγραμμα που ονομάζεται «Access KeyChain», κάνουμε κλικ στην επιλογή «Certificate Assistant» και μετά «Request a Certificate From a Certificate Authority». Μετά εισάγουμε την διεύθυνση ηλεκτρονικού ταχυδρομείου μας και επιλέγουμε την επιλογή «Saved to Disk». Πρέπει να γίνει η αποθήκευσή του γιατί θα χρειαστεί παρακάτω.
4. Έπειτα δημιουργούμε ένα πιστοποιητικό παραγωγής App Store (App Store Production Certificate), πηγαίνοντας στην ιστοσελίδα που πήγαμε και στο δεύτερο βήμα. Κάνουμε κλικ στην επιλογή «Certificates» και μετά το «+» στην πάνω δεξιά γωνία της οθόνης. Κάνουμε κλικ στην επιλογή «App Store» και μετά στο «Continue». Έπειτα μεταφορτώνουμε το Certificate Signing Request και τέλος κατεβάζουμε το πιστοποιητικό και το εγκαθιστούμε κάνοντας διπλό κλικ.
5. Μετά δημιουργούμε προφίλ παράδοσης παραγωγής (Production Provisioning Profile), πηγαίνοντας και πάλι στην ίδια ιστοσελίδα και κάνουμε κλικ στο «Provisioning Profiles» και πάνω δεξιά στην οθόνη το

«+». Πατάμε το «App Store Distribution option» και μετά το «Continue». Στη συνέχεια επιλέγουμε το αναγνωριστικό εφαρμογής (Bundle ID) από το 2^ο βήμα και πατάμε «Continue». Επιλέγουμε το αίτημα υπογραφής πιστοποιητικού (Certificate Signing Request) που αποθηκεύσαμε στο 3^ο βήμα και πατάμε «Continue» και «Generate». Κατεβάζουμε το αρχείο και το εγκαθιστούμε στον υπολογιστή.

6. Δημιουργούμε την καταχώρηση App Store (App Store Listing). Μεταβαίνουμε στο iTunes Connect (<https://itunesconnect.apple.com/>), κάνουμε σύνδεση και επιλέγουμε «My Apps» και το «+» πάνω δεξιά στην οθόνη. Επιλέγουμε «New iOS App» και στη συνέχεια επιλέγουμε το αναγνωριστικό εφαρμογής (Bundle ID) που φτιάξαμε στο 2^ο βήμα και πατάμε «Create» για να δημιουργηθεί η πρώτη έκδοση της εφαρμογής.
7. Σαν επόμενο βήμα δημιουργούμε την έκδοση της εφαρμογής. Ξεκινάμε το Xcode, ανοίγουμε το project ή τον χώρο εργασίας, ενημερώνουμε την έκδοση και δημιουργούμε τα build numbers. Ανοίγουμε τις ρυθμίσεις της κατασκευής και μεταβαίνουμε στην επιλογή «Code Signing», όπου χρησιμοποιούμε το προφίλ δημιουργίας, όπως και την ταυτότητα υπογραφής κώδικα που φτιάξαμε νωρίτερα. Στο μενού που βρίσκεται πάνω, επιλέγουμε «Generic iOS Device» ως προορισμό δημιουργίας. Κάνουμε κλικ στην επιλογή «Distribute» και συνδεόμαστε ως λογαριασμός προγραμματιστή Apple. Υποβάλουμε την εφαρμογή στο App Store και περιμένουμε για την επιβεβαίωση.
8. Μετά συμπληρώνουμε τις πληροφορίες για την πρώτη έκδοση της εφαρμογής. Καταχωρούμε τις πληροφορίες που συγκεντρώσαμε στο 1^ο βήμα, επιλέγουμε την εφαρμογή που θέλουμε να συνδέσουμε με την έκδοση, συμπληρώνουμε τις πληροφορίες τιμολόγησης και κάνουμε αποθήκευση.
9. Υποβάλουμε την έκδοση για αναθεώρηση. Επιλέγουμε τον τρόπο απελευθέρωσης της εφαρμογής και περιμένουμε να εγκριθεί.
10. Τέλος στην περίπτωση που έχουμε επιλέξει μη αυτόματη απελευθέρωση στο 9^ο βήμα, κάνουμε κλικ στην επιλογή «απελευθέρωση» όταν είμαστε έτοιμοι και περιμένουμε μέχρι η κατάσταση την έκδοσης να αποκτήσει πράσινο χρώμα (δηλαδή έτοιμο προς πώληση). Αυτή η διαδικασία μπορεί να διαρκέσει από 1 ώρα μέχρι 1 ημέρα

Windows Store:

Το κόστος για να ανεβεί μία εφαρμογή στο Windows Store είναι 20 ευρώ.

1. Αρχικά κάνουμε σύνδεση στην ιστοσελίδα (<https://dev.windows.com>) .
2. Μετά στην σελίδα που θα εμφανιστεί επιλέγουμε «Create a new app».
3. Επιλέγουμε όνομα για την εφαρμογή και εφόσον είναι διαθέσιμο (εμφανιστεί πράσινη σήμανση δίπλα στο όνομα της εφαρμογής) πατάμε «Reserve app name».
4. Στη συνέχεια υποβάλουμε τα στοιχεία της εφαρμογής επιλέγοντας «Start Your Submission».
5. Συμπληρώνουμε το κόστος που θέλουμε να πληρώσει ο χρήστης, επιλέγοντας «Pricing And Availability» που θέλει να αποκτήσει την εφαρμογή και αν θα διατίθεται για δωρεάν δοκιμή για κάποιο ορισμένο χρόνο. Μετά επιλέγουμε «Publish this app soon as it passes certification».
6. Δηλώνουμε σε ποια κατηγορία ανήκει η εφαρμογή (πχ παιχνίδι, εκπαίδευση) και την ηλικία για την οποία είναι διαθέσιμη η εφαρμογή (3+, 7+, 12+, 16+, 18+).
7. Φορτώνουμε την εφαρμογή πατώντας «browse your files» και επιλέγουμε την θέση της εφαρμογής.
8. Δίνουμε μία περιγραφή για την εφαρμογή και εισάγουμε την λειτουργία της εφαρμογής και επιπλέον στοιχεία που χρειάζονται.
9. Τέλος πατάμε την επιλογή «Submit to the Store».

(βλ. [ΒΙΒΛΙΟΓΡΑΦΙΑ \[9.1\], \[9.2\], \[9.3\]](#))

3 Ανάπτυξη Εφαρμογών σε Android.

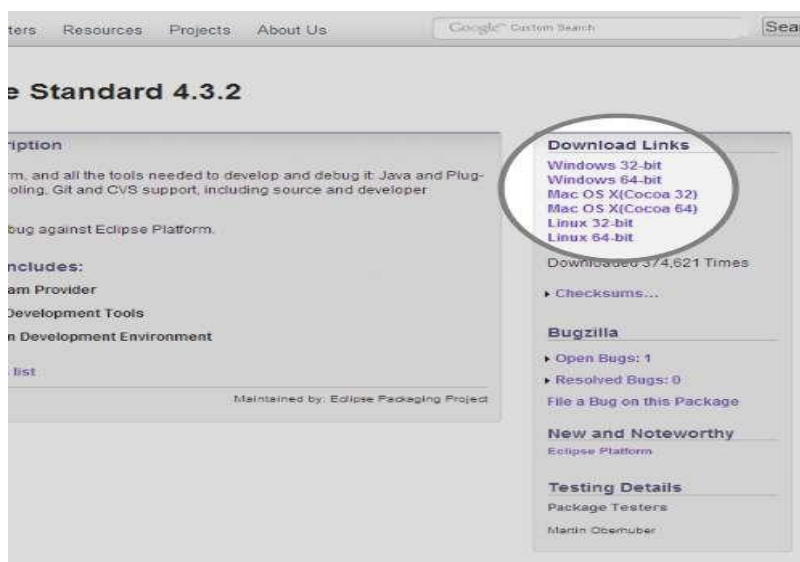
Υπάρχει τεράστια γκάμα εφαρμογών για τις Android συσκευές, που όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, γίνονται διαθέσιμες μέσω του Play Store. Τέτοιες εφαρμογές μπορούν να αναπτυχθούν από τον καθένα, αρκεί να γνωρίζει την Java

για Android. Για να αναπτύξει κανείς μία τέτοια εφαρμογή, χρειάζεται και το απαραίτητο λογισμικό, το οποίο ονομάζεται Android Plugin. Αυτό μπορεί να εγκατασταθεί σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment IDE), όπως το Eclipse, το οποίο χρησιμοποιήθηκε και για την ανάπτυξη της εφαρμογής που παρουσιάζεται σε αυτή την πτυχιακή. Σ' αυτό το κεφάλαιο αναφέρεται ο τρόπος εγκατάστασης του Eclipse IDE, αλλά και του Android Plugin μέσα σε αυτό, αλλά και κάποια παραδείγματα χρήσης του Android στο internet.

3.1 Εγκατάσταση του Android Plugin σε Eclipse.

Βήμα 1º: Απόκτηση του Eclipse IDE

Πηγαίνουμε στην ιστοσελίδα [http://www.eclipse.org/downloads/packages/eclipse-](http://www.eclipse.org/downloads/packages/eclipse-standard-432/keplersr2)



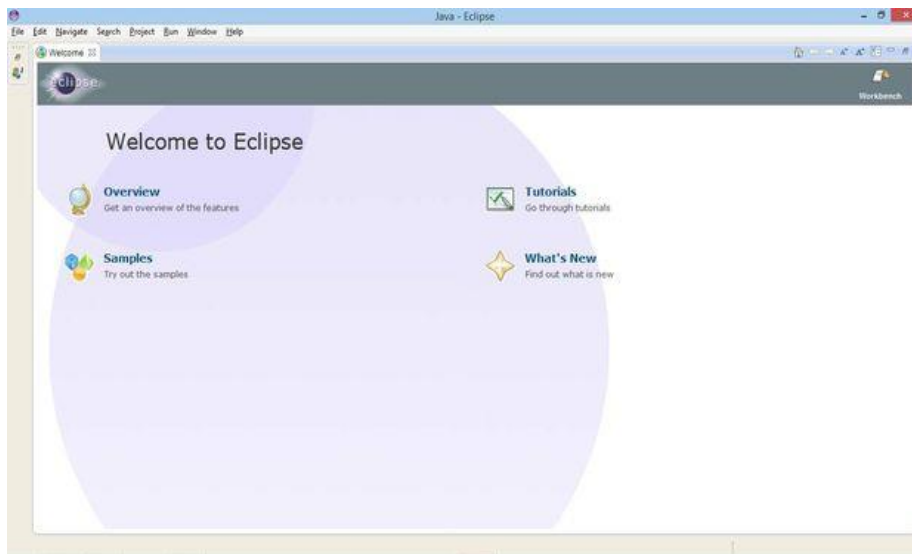
[standard-432/keplersr2](http://www.eclipse.org/downloads/packages/eclipse-standard-432/keplersr2) , για να κατεβάσουμε το Eclipse, επιλέγοντας με βάση το λογισμικό του υπολογιστή μας.

Εικόνα 3 Απόκτηση του Eclipse IDE

Βήμα 2º: Πρώτη εκτέλεση του Eclipse IDE

Το Eclipse δεν χρειάζεται εγκατάσταση, αλλά το αρχείο που κατεβάσαμε στο προηγούμενο βήμα, το οποίο είναι ένας συμπίεσμένος φάκελος που μέσα περιέχει και το

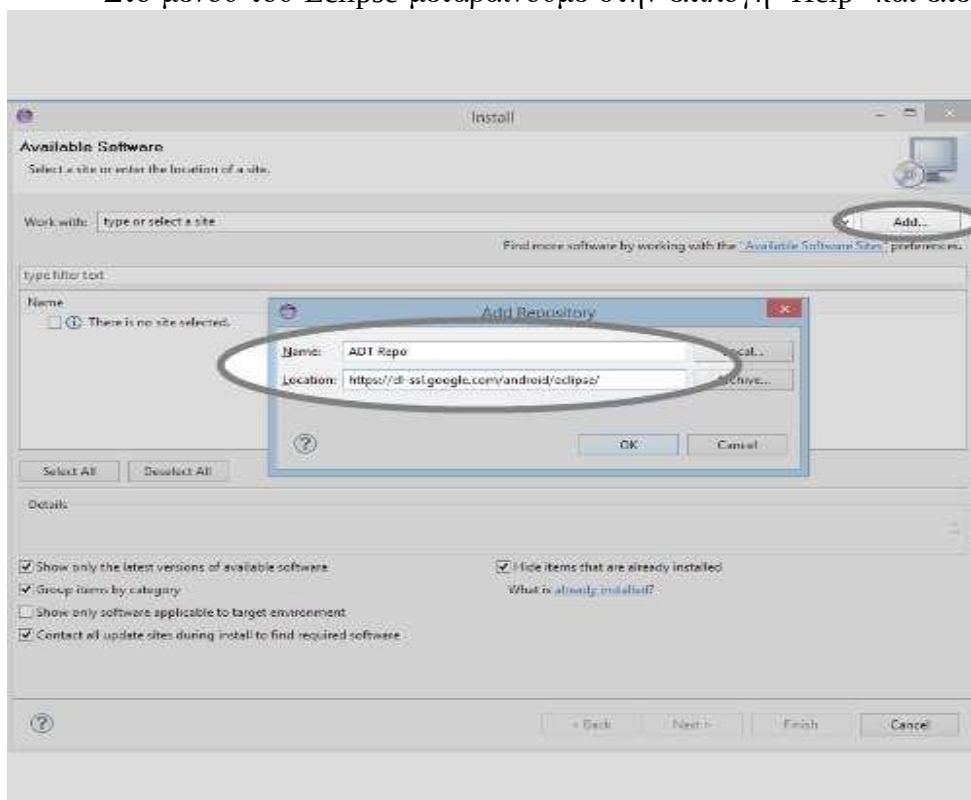
εκτελέσιμο αρχείο με το όνομα eclipse.exe. Μόλις γίνει η εκκίνηση του προγράμματος θα πρέπει να δούμε κάτι παρόμοιο με την *Εικόνα 4*.



Εικόνα 4 Πρώτη εκτέλεση του Eclipse IDE

Βήμα 3^ο: Προσθήκη και εγκατάσταση του Android Development Tools (ADT Plugin for Eclipse)

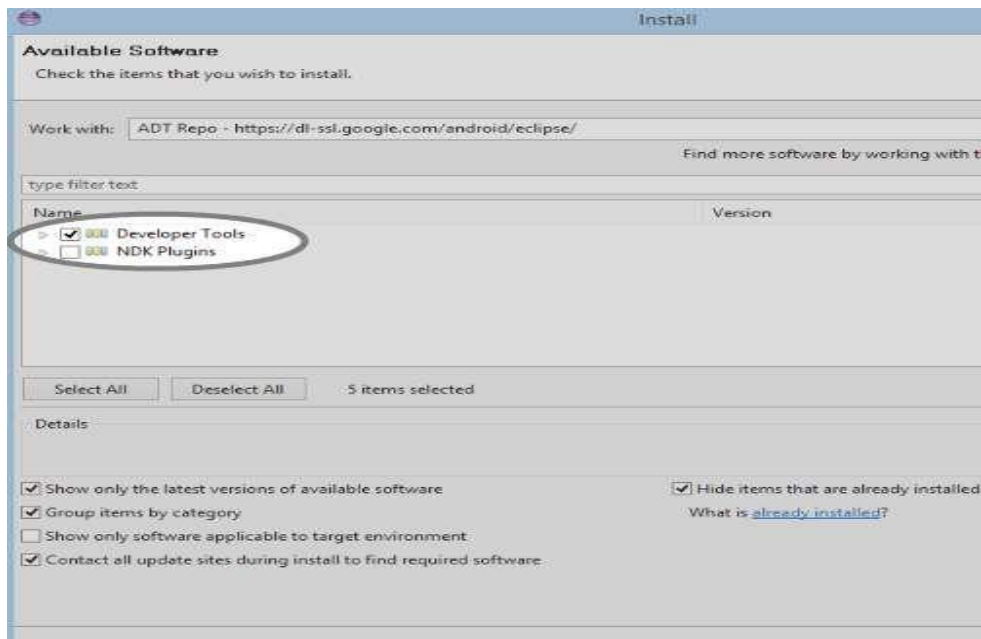
Στο μενού του Eclipse μεταβαίνουμε στην επιλογή ‘Help’ και έπειτα ‘Install New Software’.



Εικόνα 5 Προσθήκη και εγκατάσταση του Android Development Tools (ADT Plugin for Eclipse)

Στη συνέχεια πατάμε το κουμπί ‘Add’ που εμφανίζει ένα νέο παράθυρο. Δίνουμε ένα όνομα όπως π.χ ‘ADT Repo’ και στο πεδίο «location» το link [http://dl-](http://dl-ssl.google.com/android/eclipse/)

ssl.google.com/android/eclipse/ και πατάμε «OK». Τώρα πρέπει να εγκαταστήσουμε το ADT, επιλέγοντας το «Developer Tools» από την λίστα που εμφανίζεται και πατάμε ‘Next’ και δεχόμαστε του όρους αφού τους διαβάσουμε και τέλος πατάμε ‘Finish’.

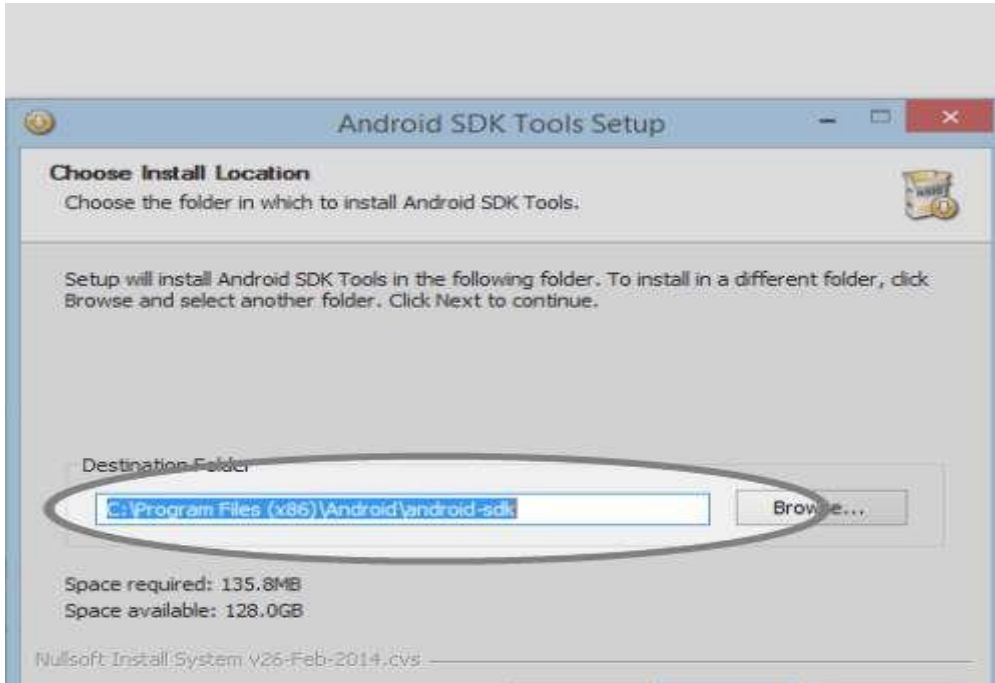


Εικόνα 6 Προσθήκη και εγκατάσταση του Android Development Tools (ADT Plugin for Eclipse)

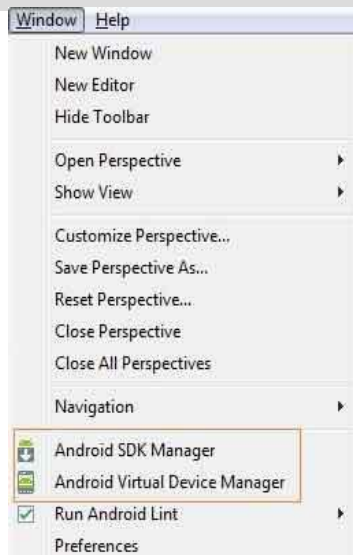
Βήμα 4º: Εγκατάσταση του Android SDK

Μετά την εγκατάσταση του ADT, το eclipse θα κάνει επανεκκίνηση και μόλις ξαναμπει στο περιβάλλον του θα μας ζητήσει να δώσουμε το μονοπάτι στο οποίο επιθυμούμε να

εγκαταστήσουμε το Android SDK ή αν θέλουμε να εγκατασταθεί στο προκαθορισμένο μονοπάτι, όπως φαίνεται και στην *Εικόνα 7*.



Εικόνα 7 Εγκατάσταση του Android SDK (1)

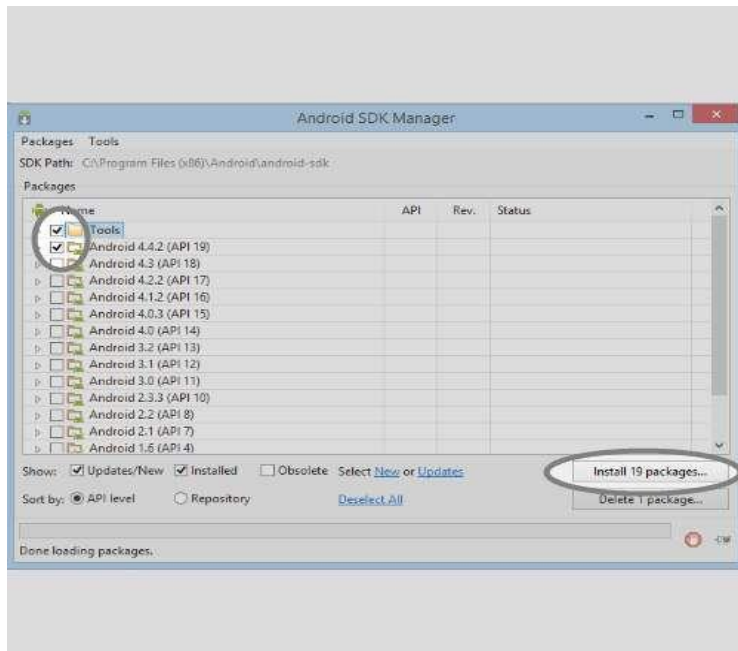


Εικόνα 8 Εγκατάσταση του Android SDK (2)

Για να ελέγξουμε αν έγινε με επιτυχία η εγκατάσταση του, πατάμε στο μενού του Eclipse «Window» και θα πρέπει να δούμε δύο επιλογές, τις «Android SDK Manager» και «Android Virtual Device Manager» όπως στην *Εικόνα 8*.

Βήμα 5º: Εγκατάσταση εκδόσεων Android (APIs) και Extras

Για να μπορούμε να προγραμματίσουμε Android εφαρμογές, θα πρέπει να εγκαταστήσουμε όποια ή όποιες εκδόσεις Android επιθυμούμε να



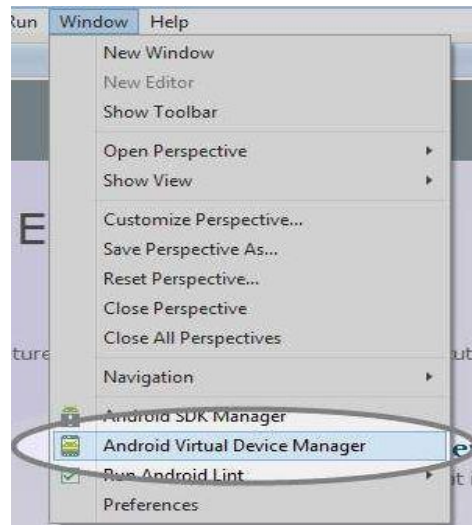
Εικόνα 9 Εγκατάσταση εκδόσεων Android (APIs) και Extras

χρησιμοποιήσουμε. Πάμε στο μενού του Eclipse και πατάμε «Window-> Android SDK Manager» και αφήνουμε επιλεγμένα από τη λίστα Tools και επιλέγουμε τις εκδόσεις που θέλουμε. Σε κάποιες εκδόσεις, η εγκατάσταση του ARM EABI v7a

System Image, είναι απαραίτητη για να μπορέσουμε να φτιάξουμε ένα Android Virtual Device, για να τρέχουμε τις εφαρμογές μας από εκεί, αν δεν διαθέτουμε Android συσκευή.

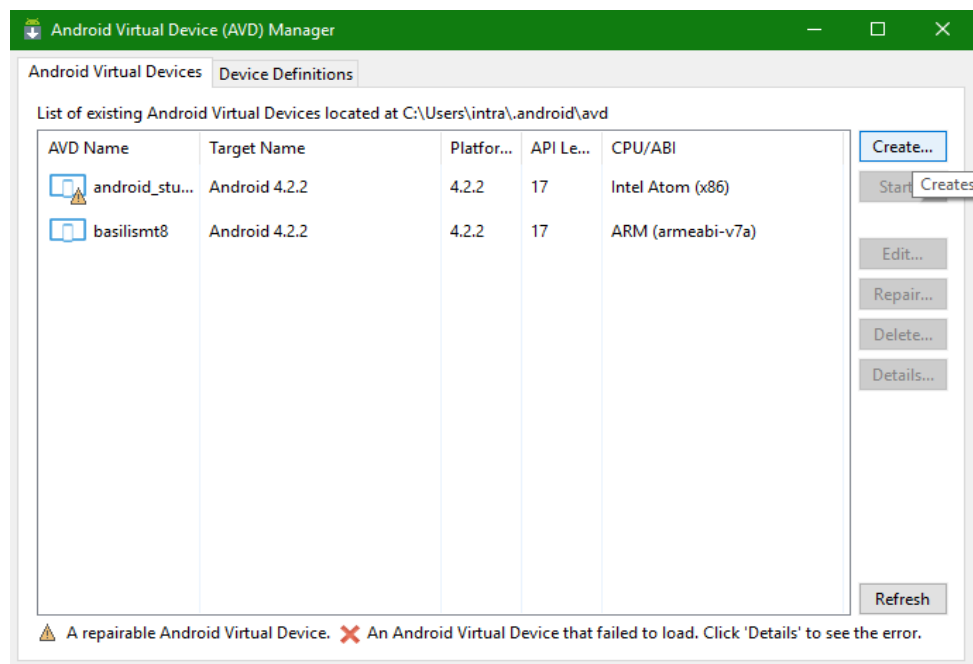
Βήμα 7ο: Δημιουργία Android Virtual Device (AVD)

Πηγαίνουμε στο μενού του Eclipse και πατάμε «Window-> Android Virtual Device Manager» όπως στην *Εικόνα 10*.



Εικόνα 10 Δημιουργία Android Virtual Device (AVD) (1)

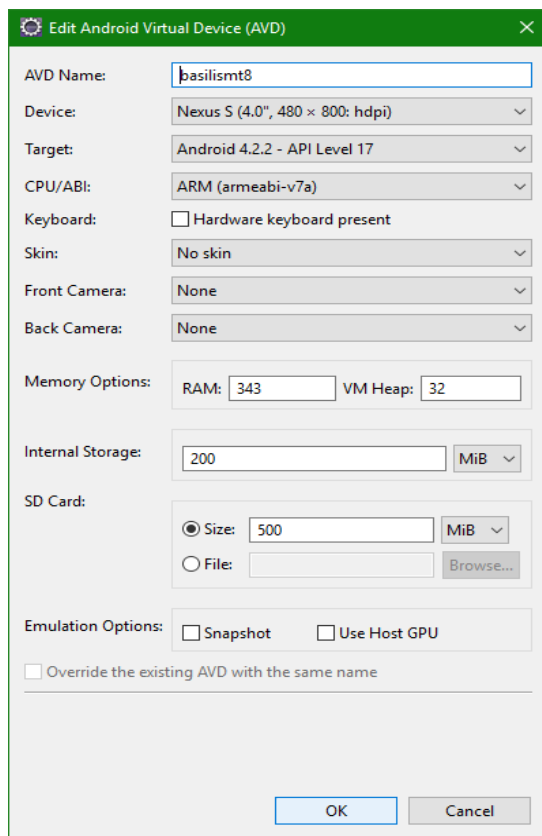
Μετά στο παράθυρο που εμφανίζεται πατάμε «Create» όπως στην *Εικόνα 11*.



Εικόνα 11 Δημιουργία Android Virtual Device (AVD) (1)

Τέλος κάνουμε τις ρυθμίσεις για το AVD, δίνοντας το επιθυμητό όνομα, ποια συσκευή θέλουμε να

προσομοιάσουμε στο Device και ποια έκδοση Android θέλουμε να έχει το AVD μας, επιλέγοντας μία από αυτές που εγκαταστήσαμε στο Βήμα 5^ο και τα υπόλοιπα όπως φαίνονται στην *Εικόνα 12* και πατάμε «OK».



Εικόνα 12 Δημιουργία Android Virtual Device (AVD) (3)

[\(βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[10.1\], \[10.2\]\)](#)

3.2 Χρήση του Android στο Internet.

Internet και android εφαρμογές πάνε μαζί. Οι περισσότερες εφαρμογές απαιτούν σύνδεση μεταξύ εφαρμογής και internet, όπως και αυτή που παρουσιάζεται σε αυτή την πτυχιακή, που για να κατεβάσει τα περιεχόμενα από τον ιστότοπο (site) πρέπει να επικοινωνήσουν δημιουργώντας μία σύνδεση. Σε αυτή την ενότητα παρουσιάζονται κάποια τέτοια μικρά παραδείγματα.

Παράδειγμα 1º: Ανάκτηση ιστοσελίδας.

```
URL url = new URL("http://www.android.com/");
URLConnection urlConnection = (URLConnection)
url.openConnection();
try {
    InputStream in = new
BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
} finally {
    urlConnection.disconnect();
}

URL url = new URL("http://www.android.com/");
URLConnection urlConnection = (URLConnection)
url.openConnection();
try {
    InputStream in = new
BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
} finally {
    urlConnection.disconnect();
}
```

Παράδειγμα 2º: Http Post Request.

```

public class HTTPPostActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        makePostRequest();
    }
    private void makePostRequest() {
        HttpClient httpClient = new DefaultHttpClient();
        // replace with your url
        HttpPost httpPost = new
HttpPost("www.example.com");
        //Post Data
        List<NameValuePair> nameValuePair = new
ArrayList<NameValuePair>(2);
        nameValuePair.add(new
BasicNameValuePair("username", "test_user"));
        nameValuePair.add(new
BasicNameValuePair("password", "123456789"));
        //Encoding POST data
        try {
            httpPost.setEntity(new
UrlEncodedFormEntity(nameValuePair));
        } catch (UnsupportedEncodingException e) {
            // log exception
            e.printStackTrace();
        }
        //making POST request.
        try {
            HttpResponse response =
httpClient.execute(httpPost);
            // write response to log
            Log.d("Http Post Response:",
response.toString());
        } catch (ClientProtocolException e) {
            // Log exception
            e.printStackTrace();
        } catch (IOException e) {
            // Log exception
            e.printStackTrace();
        }
    }
}

```

Παράδειγμα 3^ο: Http Get Request.

```

public class HTTPGETActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        makeGetRequest();
    }
    private void makeGetRequest() {
        HttpClient client = new DefaultHttpClient();
        HttpGet request = new
HttpGet("http://www.example.com");
// replace with your url

        HttpResponse response;
        try {
            response = client.execute(request);
            Log.d("Response of GET request",
response.toString());
        } catch (ClientProtocolException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

(βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ [11])

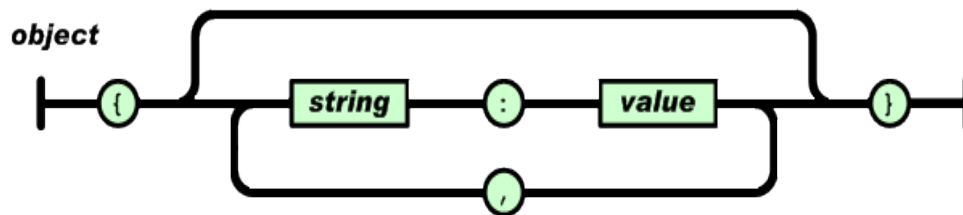
3.3 Αποθήκευση δεδομένων σε Android προγράμματα.

Στις εφαρμογές για android κινητές συσκευές, για αποθήκευση δεδομένων μπορούν να χρησιμοποιηθούν τύποι αρχείων, όπως τα Json, Xml, Txt. Επίσης εκτός από αρχεία μπορεί να χρησιμοποιηθεί και βάση δεδομένων, η SQLite. Στην εφαρμογή που παρουσιάζεται σε αυτή την πτυχιακή εργασία, χρησιμοποιήθηκαν τα Json αρχεία για την λήψη δεδομένων από την ιστοσελίδα του τμήματος νοσηλευτικής, του Τ.Ε.Ι Ηπείρου. Ακολουθεί μία περιληπτική ανάλυση τους.

Json: Τα Json (JavaScript Object Notation), είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο στο να γραφτούν και να διαβαστούν από τον άνθρωπο και από τις μηχανές είναι εύκολο να το αναλύσουν (parse) και να το παράγουν (generate). Είναι ένα πρότυπο κειμένου τελείως ανεξάρτητο από γλώσσες προγραμματισμού, αλλά χρησιμοποιεί πρακτικές οι οποίες είναι γνωστές στους προγραμματιστές των γλωσσών C, C++, C#, Java,

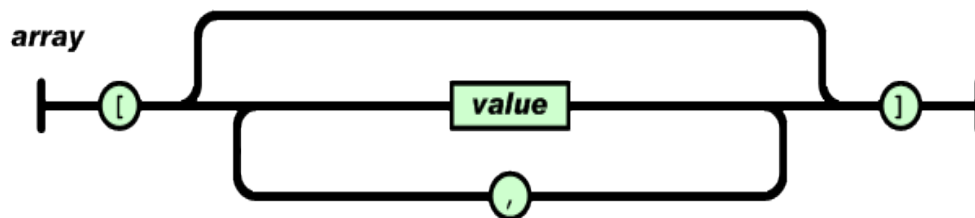
JavaScript, Perl, Python και πολλών άλλων. Αυτό ο τύπος αρχείου χρησιμοποιείται πάρα πολύ σε εφαρμογές διαδικτύου. Είναι χτισμένο σε δύο μορφές που είναι οι εξής: **1)** Μία συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash, λίστα κλειδιών ή associative πίνακα. **2)** Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας, διάνυσμα, λίστα ή ακολουθία.

Ένα αντικείμενο (object) είναι ένα άτακτο σύνολο από ζευγάρια ονομάτων/τιμών. Ένα αντικείμενο ξεκινάει με αριστερό άγκιστρο ({) και τελειώνει με δεξί άγκιστρο (}). Κάθε όνομα ακολουθείται από άνω κάτω τελεία (:) και τα ζευγάρια ονόματος/τιμής χωρίζονται από κόμμα (,).



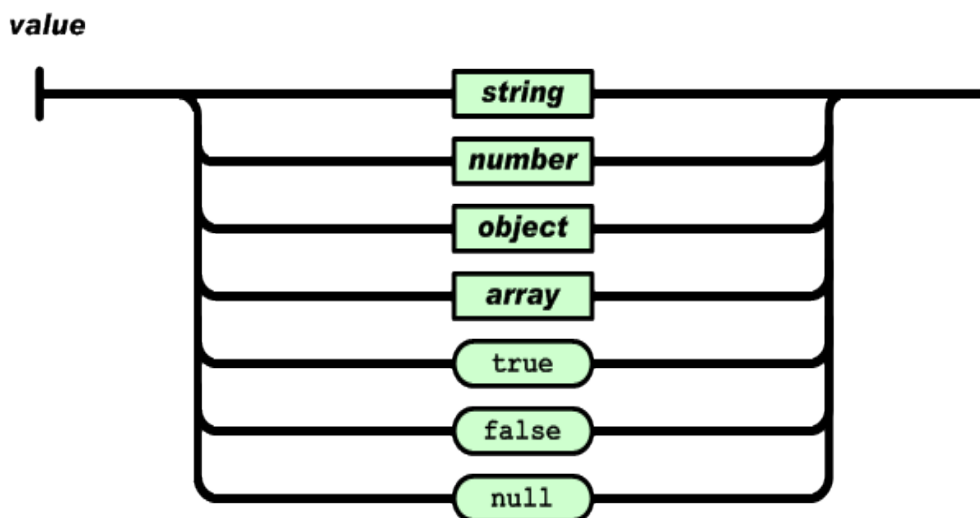
Εικόνα 13 Json objects

Ένας πίνακας (array) είναι μια συλλογή από τιμές σε σειρά. Ένας πίνακας ξεκινάει με αριστερή αγκύλη ([) και τελειώνει με δεξιά αγκύλη (]). Οι τιμές χωρίζονται με κόμμα (,).



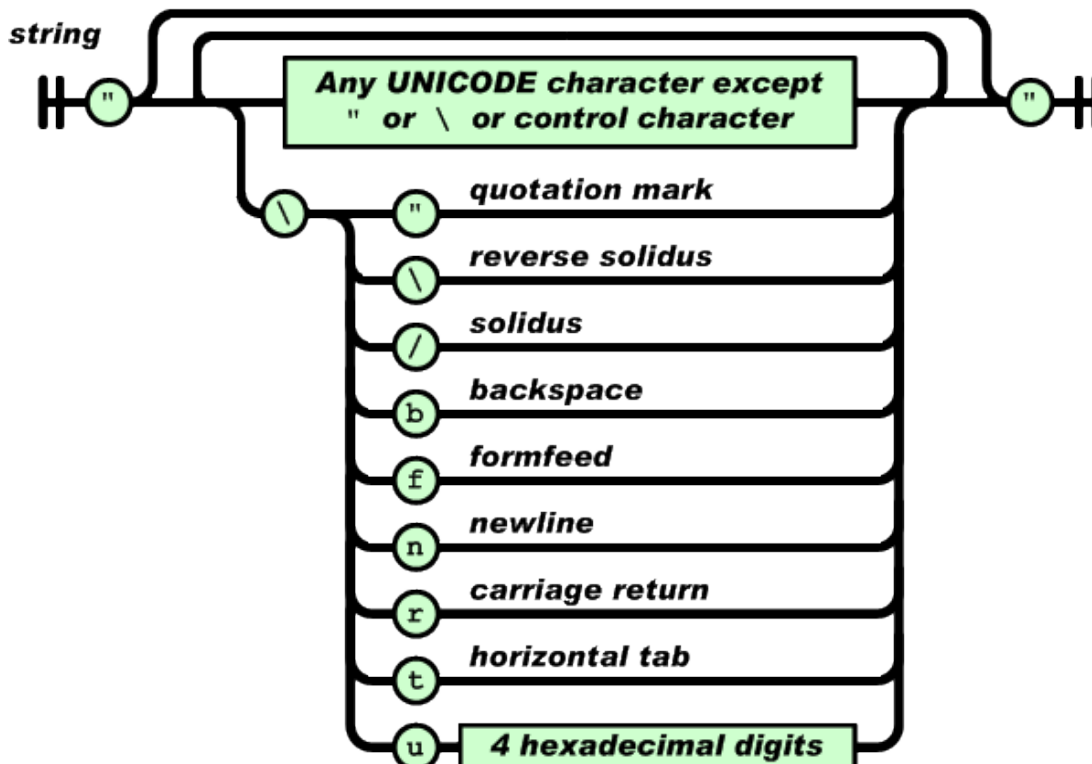
Εικόνα 14 Json Arrays

Μία τιμή μπορεί να είναι string μέσα σε διπλά quotes (“ ”), ή αριθμός, ή true ή false ή null, ή αντικείμενο (object) ή πίνακας (array). Αυτές οι τιμές μπορεί να είναι και ανακατεμένες.



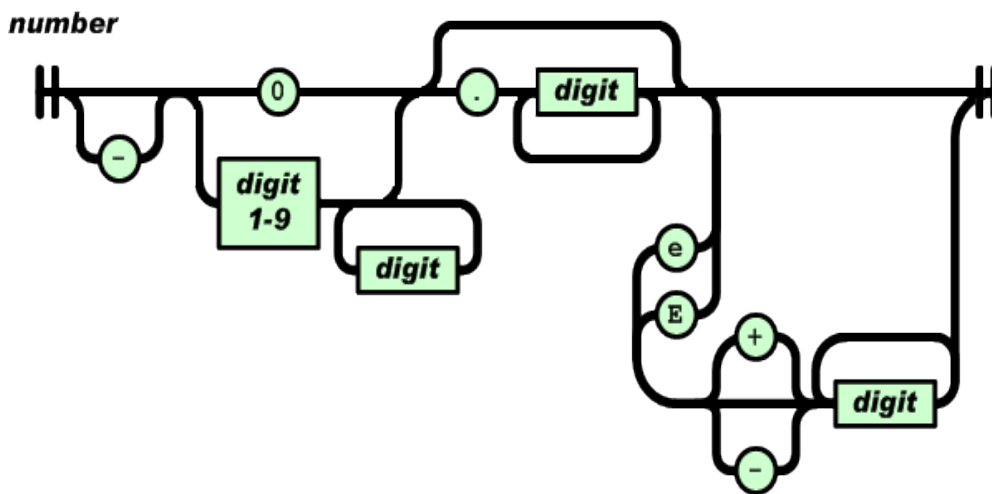
Εικόνα 15 Json Values

Ένα string είναι μια συλλογή από μηδέν ή περισσότερους Unicode χαρακτήρες, μέσα σε διπλά quotes (“ ”), χρησιμοποιώντας αντίστροφους κάθετους (\) για διαφυγή. Ένας χαρακτήρας αντιπροσωπεύεται ως ένας μονός χαρακτήρας string. Ένα string μοιάζει πολύ σαν ένα C ή Java string.



Εικόνα 16 Json Strings

Ένας αριθμός μοιάζει πάρα πολύ με ένα C ή Java αριθμό, με την διαφορά πως τα οκταδικά και δεκαεξαδικά συστήματα δεν χρησιμοποιούνται.



Εικόνα 17 Json Numbers

Xml: Τα Xml (eXtensible Markup Language) χρησιμοποιούνται περισσότερο στο παρελθόν και πλέον χρησιμοποιούνται για την μορφοποίηση εφαρμογών. Είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση των Xml εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν, για παράδειγμα, στις υπηρεσίες ιστού.

Τα βασικά στοιχεία που συναντώνται στην καθημερινή χρήση τους αναφέρονται παρακάτω:

Χαρακτήρας Unicode: Εξ ορισμού ένα κείμενο Xml είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανιστεί σε ένα κείμενο Xml.

Επεξεργαστής και Εφαρμογή: Είναι το λογισμικό που απαρτίζουν ένα κείμενο Xml. Αποτελούν είτε τη *σήμανση*, είτε το *περιεχόμενο* του. Η σήμανση και το περιεχόμενο μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με τον χαρακτήρα (<) και καταλήγουν στο χαρακτήρα (>), είτε ξεκινούν με το χαρακτήρα (

&) και καταλήγουν στο χαρακτήρα (;). Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση αποτελούν το περιεχόμενο ενός κειμένου Xml.

Ετικέτα: Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα (<) και καταλήγει στο χαρακτήρα (>). Υπάρχουν τρία είδη ετικέτας: *ετικέτες αρχής*, για παράδειγμα <section> , *ετικέτες τέλους* για παράδειγμα </section> και *ετικέτες χωρίς περιεχόμενο* για παράδειγμα <line-break/>.

Στοιχείο: Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα αρχής και καταλήγει σε μία ετικέτα τέλους, είτε αποτελείται μόνο από μία ετικέτα χωρίς περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας αρχής και μιας ετικέτας τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

Χαρακτηριστικό: Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι *όνομα/τιμή*, το οποίο υπάρχει μέσα σε μία ετικέτα αρχής ή σε μία ετικέτα χωρίς περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο *img* έχει δύο χαρακτηριστικά, τα *src* και *alt*: . Ένα άλλο παράδειγμα θα ήταν το <step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση Xml: Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα: <?xml version="1.0" encoding="UTF-8"?>.

Ακολουθεί ένα πλήρες κείμενο Xml, που χρησιμοποιεί όλες τις παραπάνω έννοιες.

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>
  
  <caption>This is Raphael's "Foligno" Madonna, painted in
  <date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Txt: Αυτός ο τύπος αρχείου χρησιμοποιείται περισσότερο για αποθήκευση κειμένου, παρά για αποθήκευση δεδομένων. Γενικά είναι ένα τυπικό έγγραφο κειμένου που

περιέχει μη μορφοποιημένο κείμενο. Αναγνωρίζεται από οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου.

SqlLite: Η SqlLite είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Σε αντίθεση με άλλα συστήματα διαχείρισης βάσης δεδομένων, η SqlLite δεν είναι μία ξεχωριστή διεργασία που προσπελάζεται από μία εφαρμογή πελάτη, αλλά ένα ενσωματωμένο μέρος της εφαρμογής. Έχει δυνατότητες όπως με τις βάσεις δεδομένων SQL, δηλαδή μπορούν να αποθηκευτούν δεδομένα και να επεξεργαστούν με ερωτήματα insert για την εισαγωγή δεδομένων, delete για της διαγραφή δεδομένων και select για την εμφάνιση δεδομένων. ([βλ. ΒΙΒΛΙΟΓΡΑΦΙΑ \[12.1\],\[12.2\]\[12.3\]\[1.4\]](#))

4 Η Προτεινόμενη Εφαρμογή

Σε αυτό το κεφάλαιο γίνεται η ανάλυση του τρόπου λειτουργίας της εφαρμογής. Παρουσιάζονται ο ιστότοπος του τμήματος νοσηλευτικής, τα αρχεία κώδικα (Web Services) που χρησιμοποιήθηκαν για να γίνει η σύνδεση της εφαρμογής με τον ιστότοπο, πως λειτουργεί η εφαρμογή με την βοήθεια εικόνων (Screenshots) και επεξηγηματικού κειμένου. Τέλος αναφέρονται τυχόν μελλοντικές επεκτάσεις της εφαρμογής, δηλαδή σημεία που μπορεί να βελτιωθούν, είτε στο γραφικό περιβάλλον είτε καινούριες δυνατότητες που μπορεί να προστεθούν.

4.1 Η ιστοσελίδα της εφαρμογής

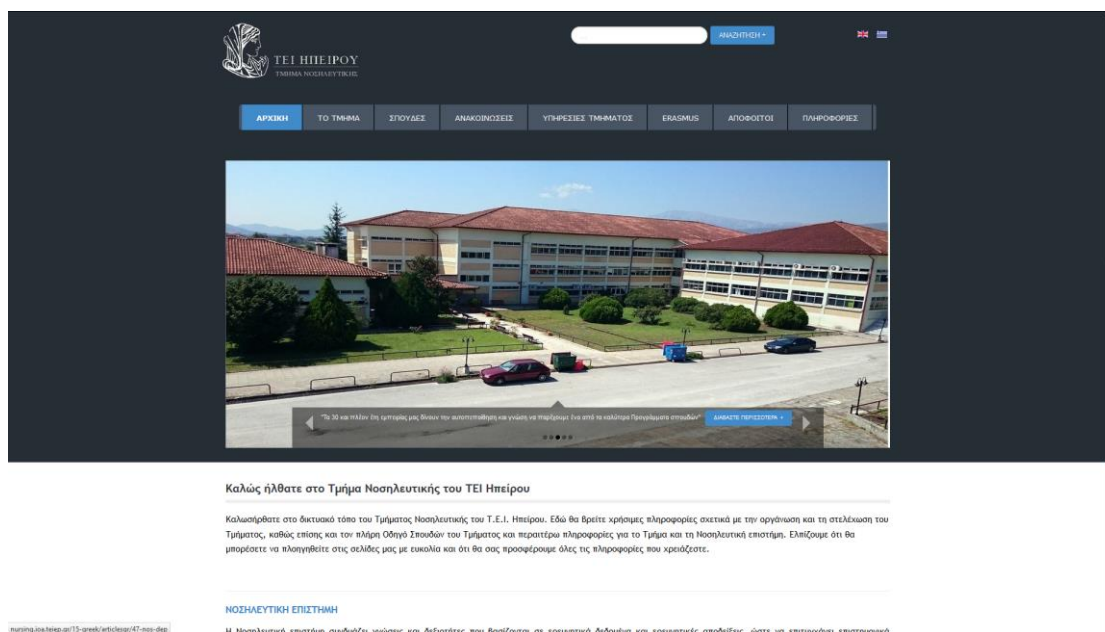
Όπως έχει αναφερθεί και παραπάνω η εφαρμογή που παρουσιάζεται σε αυτή την πτυχιακή, διαχειρίζεται το περιεχόμενο του ιστότοπου (site), του τμήματος νοσηλευτικής του Τ.Ε.Ι Ηπείρου (<http://nursing.ioa.teiep.gr/>). Για να μπορέσει η εφαρμογή να γίνει κατανοητή, θα πρέπει να παρουσιαστεί και η βασική δομή του ιστότοπου αυτού.

Η εφαρμογή προσφέρει τις ανακοινώσεις του τμήματος νοσηλευτικής, πληροφορίες και δυνατότητα επικοινωνίας με το προσωπικό του τμήματος αλλά και με τους καθηγητές

μέσω email ή τηλεφώνου, βασικές πληροφορίες για το πρόγραμμα σπουδών και τέλος κάποιους χρήσιμους συνδέσμους (link), για διάφορες υπηρεσίες για τους φοιτητές.

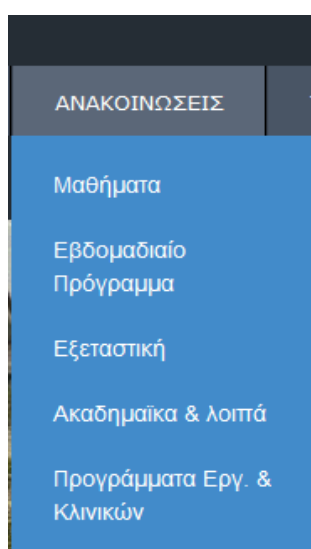
Ακολουθούν κάποιες εικόνες από την ιστοσελίδα.

Αρχική σελίδα:



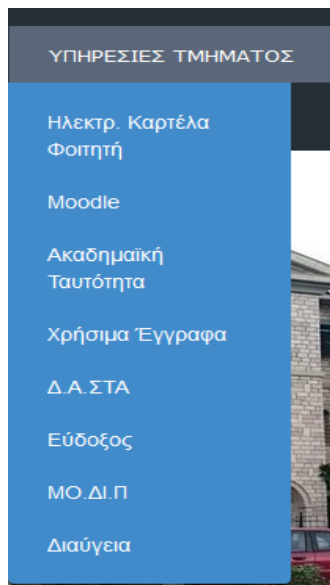
Εικόνα 18 Αρχική σελίδα

Κατηγορίες ανακοινώσεων:



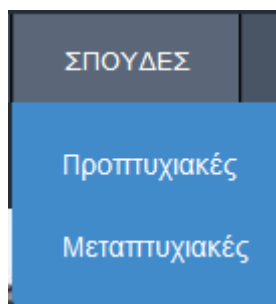
Εικόνα 19 Κατηγορίες ανακοινώσεων

Υπηρεσίες τμήματος:



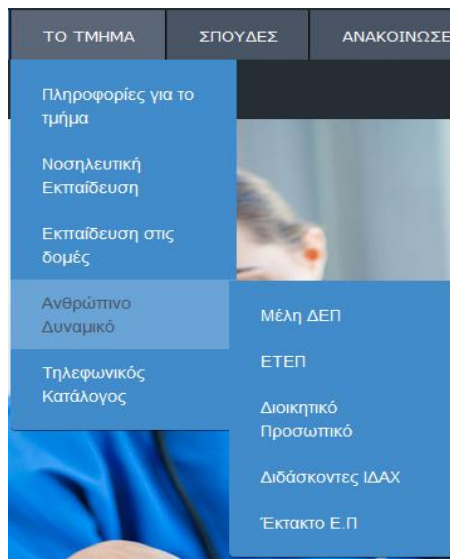
Εικόνα 20 Υπηρεσίες τμήματος

Σπουδές:



Εικόνα 21 Σπουδές

Ανθρώπινο δυναμικό:



Εικόνα 22 Ανθρώπινο δυναμικό

4.2 Οι χρησιμοποιούμενες web services

Για να γίνει η σύνδεση της εφαρμογής με την ιστοσελίδα χρησιμοποιήθηκαν κάποια αρχεία κώδικα σε γλώσσα PHP μαζί με SQL ερωτήματα, έτσι ώστε να μπορεί η εφαρμογή να παίρνει τα δεδομένα που χρειάζεται από τη βάση δεδομένων της ιστοσελίδας όπως οι ανακοινώσεις. Αυτά τοποθετήθηκαν στον φάκελο της ιστοσελίδας μέσα στον Server που την φιλοξενεί.

Παρακάτω παρουσιάζονται οι web services μαζί με σύντομη εξήγηση.

Αρχικά δημιουργήθηκε ένα αρχείο κώδικα με όνομα settings.php που περιέχει τα στοιχεία σύνδεσης της βάσης δεδομένων, που καλείται σε όλα τα υπόλοιπα web services για να μην πληκτρολογούνται οι ίδιες πληροφορίες συνέχεια.

Στην συνέχεια ακολουθούν με την σειρά τα αρχεία με όνομα ana_cat.php, ana_title.php, prosopiko.php, prosopiko_intr.php, και article.php. Όλα τα αρχεία έχουν ίδια δομή. Αρχικά καλούνε μέσω της συνάρτησης **importSettings** το αρχείο

settings.php, για να μπορούν στην συνέχεια μέσα στην κλάση API και στην συνάρτηση δημιουργίας να κάνουν την σύνδεση με την βάση δεδομένων, με βάση τα στοιχεία από το

```
<?php
    $IPADD = '***';
    $USER = '***';
    $PASS = '***';
    $DBASE = '***';
?>
```

```
<?php
    $IPADD = '***';
    $USER = '***';
    $PASS = '***';
    $DBASE = '***';
?>
```


αρχείο. Επίσης υπάρχει και μία συνάρτηση καταστροφής που τερματίζει την σύνδεση με την βάση δεδομένων.

1. **ana_cat.php** : Αυτό το αρχείο μέσω της συνάρτησης **getAll** παίρνει από την βάση δεδομένων της κατηγορίες των ανακοινώσεων όπως είδαμε παραπάνω, για να εμφανίζονται στην εφαρμογή, έτσι ώστε ο χρήστης να επιλέγει να δει ανακοινώσεις ανά κατηγορία όπως είναι και στην ιστοσελίδα ακριβώς.
2. **ana_title.php** : Αυτό το αρχείο μέσω της συνάρτησης **getOne** παίρνει από την βάση δεδομένων της ανακοινώσεις με βάση την κατηγορία τους που είναι το όρισμα αυτής της συνάρτησης που στέλνεται από την εφαρμογή σε αυτό το αρχείο κώδικα μέσω του αρχείου ana_cat.php.
3. **prosopiko.php** : Εδώ μέσω της εφαρμογής **getAll** παίρνει από τη βάση δεδομένων όλες της κατηγορίες προσωπικού του τμήματος νοσηλευτικής, για να μπορεί ο χρήστης να δει στοιχεία επικοινωνίας. Επίσης αυτή η συνάρτηση συνδυάζεται με την **getOne** γιατί ένα πεδίο που τραβάει από τη βάση δεδομένων είναι το link, που αυτό χρησιμοποιείται σαν όρισμα στην **getOne** για να φιλτράρει το αποτέλεσμα του ερωτήματος και να τραβήξει το συγκριμένο άρθρο με βάση το link.
4. **prosopiko_intr.php** : Τώρα σε αυτό το αρχείο η **getOne** έχει σαν όρισμα έναν ακέραιο αριθμό που αποστέλλεται από την εφαρμογή, μέσω της επεξεργασίας του αποτελέσματος της συνάρτησης **getOne** του αρχείου prosopiko.php, για να εμφανίσει τις πληροφορίες επικοινωνίας με βάση την κατηγορία του προσωπικού.
5. **article.php** : Αυτό το αρχείο μέσω της συνάρτησης **getAll** παίρνει από τη βάση δεδομένων το πρόγραμμα σπουδών όπως εμφανίζεται στην ιστοσελίδα.

Ακολουθούν τα αρχεία κώδικα με τη σειρά που παρουσιάστηκαν:

ana_cat.php:

```
<?php
function importSettings () {
    return 'settings.php';
}
include_once(importSettings());
```

```

class API
{
    private $mysqli;
    private function __construct(){
        global $IPADD, $USER, $PASS, $DBASE;
        $this->mysqli = @new mysqli($IPADD, $USER, $PASS, $DBASE);
        if($this->mysqli->connect_errno)
            die(json_encode("Could not connect to database error
#{ $this->mysqli->connect_errno}"));
        if(!$this->mysqli->set_charset("utf8"))
            die(json_encode("There was a database error while
setting charset to utf8 #{ $this->mysqli->errno}"));
    }
    public function __destruct(){
        if(!$this->mysqli->connect_errno)
            $this->mysqli->close();
    }

    public function getAll(){
        if($result = $this->mysqli->query("SELECT id,title FROM
i3rt7_categories WHERE parent_id=21")){
            $result2 = array();
            while ($row = $result->fetch_assoc()){
                $result2[] = $row;
            }
            echo json_encode(array('anakoinoseis' => $result2));
            $result->close();
        } else {
            die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno}"));
        }
    }

    private static $_instance = null;
    public static function getInstance(){
        if(!isset(self::$_instance))
            self::$_instance = new self();
        return self::$_instance;
    }
}

if(!empty($_GET["call"]))

```

```

{
    switch(strtolower($_GET["call"])){
        case "all":{
            API::getInstance()->getAll();
            break;
        }
        case "one":{
            if(!empty($_GET["id"])){
                API::getInstance()->getOne($_GET["id"]);
            }
            break;
        }
    }
}
?>

```

ana_title.php:

```

<?php
function importSettings() {
    return 'settings.php';
}

include_once(importSettings());

class API
{
    private $mysqli;
    private function __construct(){
        global $IPADD, $USER, $PASS, $DBASE;
        $this->mysqli = @new mysqli($IPADD, $USER, $PASS, $DBASE);
        if($this->mysqli->connect_errno)
            die(json_encode("Could not connect to database error
#{ $this->mysqli->connect_errno }"));
        if(!$this->mysqli->set_charset("utf8"))
            die(json_encode("There was a database error while
setting charset to utf8 #{ $this->mysqli->errno }"));
    }
    public function __destruct(){
        if(!$this->mysqli->connect_errno)
            $this->mysqli->close();
    }
}

```

```

        public function getOne($catid) {
            if($statement = $this->mysqli->prepare("SELECT
title,publish_up,introtext FROM i3rt7_content WHERE catid = ? AND state =
1 ORDER BY publish_up DESC")) {
                $statement->bind_param("i", $catid);
                $statement->execute();
                $statement->bind_result($title,$publish_up,$introtext);
                $result = array();
                while($statement->fetch()) {
                    $result [] = array(
                        "title" => $title,
                        "publish_up" => $publish_up,
                        "introtext" => $introtext,
                    );
                }
                echo json_encode(array('title' => $result));
            }
            else{
                die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno }"));
            }
        }

        private static $_instance = null;
        public static function getInstance() {
            if(!isset(self::$_instance))
                self::$_instance = new self();
            return self::$_instance;
        }
    }

    if(!empty($_GET["call"]))
    {
        switch(strtolower($_GET["call"])) {
            case "all":{
                API::getInstance()->getAll();
                break;
            }
            case "one":{
                if(!empty($_GET["catid"])) {
                    API::getInstance()->getOne($_GET["catid"]);
                }
            }
        }
    }
}

```

```

        }
        break;
    }
}
}
?>

```

prosopiko.php:

```

<?php
function importSettings() {
    return 'settings.php';
}

include_once(importSettings());

class API
{
    private $mysqli;

    private function __construct() {
        global $IPADD, $USER, $PASS, $DBASE;
        $this->mysqli = @new mysqli($IPADD, $USER, $PASS, $DBASE);
        if($this->mysqli->connect_errno)
            die(json_encode("Could not connect to database error
#{ $this->mysqli->connect_errno }"));
        if(!$this->mysqli->set_charset("utf8"))
            die(json_encode("There was a database error while
setting charset to utf8 #{ $this->mysqli->errno }"));
    }

    public function __destruct() {
        if(!$this->mysqli->connect_errno)
            $this->mysqli->close();
    }

    public function getAll() {
        if($result = $this->mysqli->query("SELECT
id,title,alias,link,type FROM i3rt7_menu WHERE parent_id=225")) {
            $result2 = array();
            while ($row = $result->fetch_assoc()) {
                $result2[] = $row;
            }
            echo json_encode(array('prosopiko' => $result2));
        }
    }
}

```

```

        $result->close();
    } else {
        die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno }"));
    }
}

public function getOne($plinkid) {
    if($statement = $this->mysqli->prepare("SELECT link FROM
i3rt7_menu WHERE link=?")) {
        $statement->bind_param("i", $plinkid);
        $statement->execute();
        $statement->bind_result($link);
        $result = array();
        while($statement->fetch()) {
            $result [] = array(
                "link" => $link,
            );
        }
        echo json_encode(array('prosintr' => $result));
    }
    else{
        die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno }"));
    }
}

private static $_instance = null;
public static function getInstance() {
    if(!isset(self::$_instance))
        self::$_instance = new self();
    return self::$_instance;
}

}

if(!empty($_GET["call"]))
{
    switch(strtolower($_GET["call"])) {
        case "all": {
            API::getInstance()->getAll();
            break;
        }
    }
}

```

```

        case "one":{
            if(!empty($_GET["plinkid"])){
                API::getInstance()->getOne($_GET["plinkid"]);
            }
            break;
        }
    }
}
?>

```

prosopiko_intr.php:

```

<?php
function importSettings() {
    return 'settings.php';
}

include_once(importSettings());

class API
{
    private $mysqli;
    private function __construct(){
        global $IPADD, $USER, $PASS, $DBASE;
        $this->mysqli = @new mysqli($IPADD, $USER, $PASS, $DBASE);
        if($this->mysqli->connect_errno)
            die(json_encode("Could not connect to database error
#{ $this->mysqli->connect_errno}"));
        if(!$this->mysqli->set_charset("utf8"))
            die(json_encode("There was a database error while
setting charset to utf8 #{ $this->mysqli->errno}"));
    }
    public function __destruct(){
        if(!$this->mysqli->connect_errno)
            $this->mysqli->close();
    }

    public function getOne($plinkid){
        if($statement = $this->mysqli->prepare("SELECT introtext FROM
i3rt7_content WHERE id = ?")){
            $statement->bind_param("i", $plinkid);
            $statement->execute();

```

```

        $statement->bind_result($introtext);
        $result = array();
        while($statement->fetch()){
            $result [] = array(
                "introtext" => $introtext,
            );
        }
        echo json_encode(array('prosintr' => $result));
    }
    else{
        die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno }"));
    }
}

private static $_instance = null;
public static function getInstance(){
    if(!isset(self::$_instance))
        self::$_instance = new self();
    return self::$_instance;
}
}

if(!empty($_GET["call"]))
{
    switch(strtolower($_GET["call"])){
        case "all":{
            API::getInstance()->getAll();
            break;
        }
        case "one":{
            if(!empty($_GET["plinkid"])){
                API::getInstance()->getOne($_GET["plinkid"]);
            }
            break;
        }
    }
}
}
?>

```

article.php:


```

<?php
function importSettings() {
    return 'settings.php';
}

include_once(importSettings());

class API
{
    private $mysqli;
    private function __construct(){
        global $IPADD, $USER, $PASS, $DBASE;
        $this->mysqli = @new mysqli($IPADD, $USER, $PASS, $DBASE);
        if($this->mysqli->connect_errno)
            die(json_encode("Could not connect to database error
#{ $this->mysqli->connect_errno}"));
        if(!$this->mysqli->set_charset("utf8"))
            die(json_encode("There was a database error while
setting charset to utf8 #{ $this->mysqli->errno}"));
    }

    public function __destruct(){
        if(!$this->mysqli->connect_errno)
            $this->mysqli->close();
    }

    public function getAll(){
        if($result = $this->mysqli->query("SELECT
title,publish_up,introtext FROM i3rt7_content WHERE title='Πρόγραμμα
Σπουδών'")){
            $result2 = array();
            while ($row = $result->fetch_assoc()){
                $result2[] = $row;
            }
            echo json_encode(array('article' => $result2));
            $result->close();
        } else {
            die(json_encode("There was a database error please try
again later. #{ $this->mysqli->errno}"));
        }
    }

    private static $_instance = null;

```

```

        public static function getInstance() {
            if(!isset(self::$_instance))
                self::$_instance = new self();
            return self::$_instance;
        }
    }

    if(!empty($_GET["call"]))
    {
        switch(strtolower($_GET["call"])){
            case "all":{
                API::getInstance()->getAll();
                break;
            }
            case "one":{
                if(!empty($_GET["catid"])){
                    API::getInstance()->getOne($_GET["catid"]);

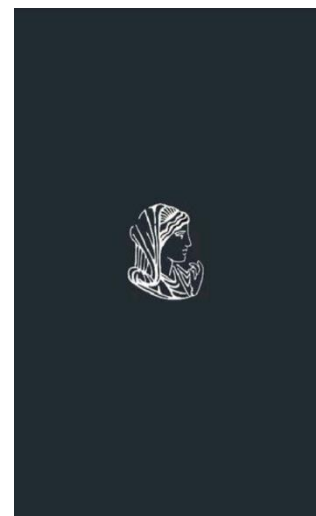
                }
                break;
            }
        }
    }
}
?>

```

4.3 Παράδειγμα χρήσης της εφαρμογής.

Ακολουθεί η παρουσίαση της λειτουργίας της εφαρμογής, με την βοήθεια screenshots.

Αρχικά στην πρώτη επαφή του χρήστη με την εφαρμογή (app), θα δει μία οθόνη που εμφανίζεται με το logo της εφαρμογής (εικόνα 23). Με την ορολογία του προγραμματισμού σε smartphones, ονομάζεται splash screen. Αυτή η οθόνη εμφανίζεται μέχρι η εφαρμογή να φορτώσει τα απαραίτητα δεδομένα για τη λειτουργία της, όπως για παράδειγμα με την εφαρμογή του Facebook, που εμφανίζει μία παρόμοια οθόνη με το logo του.



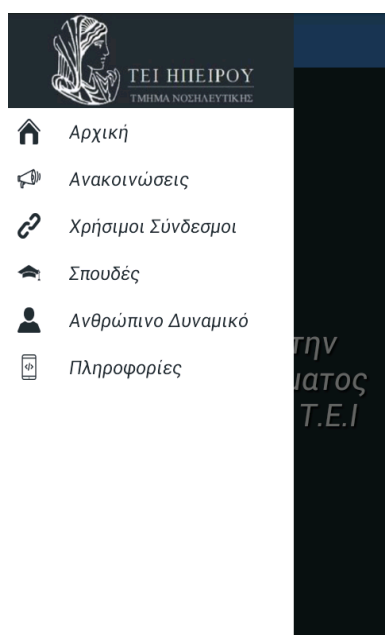
Εικόνα 23 Splash Screen

Όταν λοιπόν η εφαρμογή ανοίξει, εμφανίζεται η «Αρχική» σελίδα της εφαρμογής (εικόνα 24), που καλωσορίζει τον χρήστη στην εφαρμογή.



Εικόνα 24 Αρχική σελίδα εφαρμογής

Το μενού της είναι σε μορφή συρταριού «Navigation Drawer» (εικόνα 25), που ανοίγει είτε πατώντας το εικονίδιο με τις τρεις οριζόντιες γραμμές, είτε σέρνοντας την οθόνη

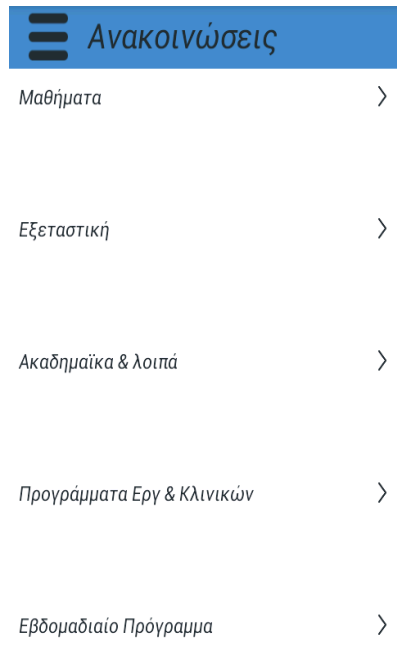


Εικόνα 25 Μενού εφαρμογής

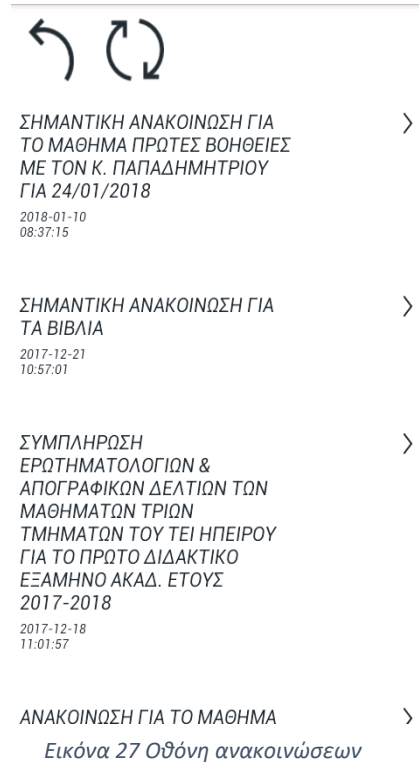
από τέρμα αριστερά προς τα δεξιά.

Η επιλογή «Ανακοινώσεις», εμφανίζει μια λίστα με τις κατηγορίες αυτών και επιλέγοντας μία, εμφανίζεται μία οθόνη της εφαρμογής με την λίστα των ανακοινώσεων της

κάθε κατηγορίας, αλλά και δύο κουμπιά, το «πίσω» και η «ανανέωση της σελίδας» (εικόνες 26,27).

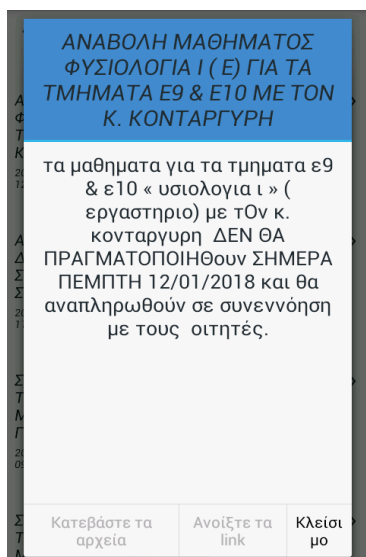


Εικόνα 26 Κατηγορίες ανακοινώσεων



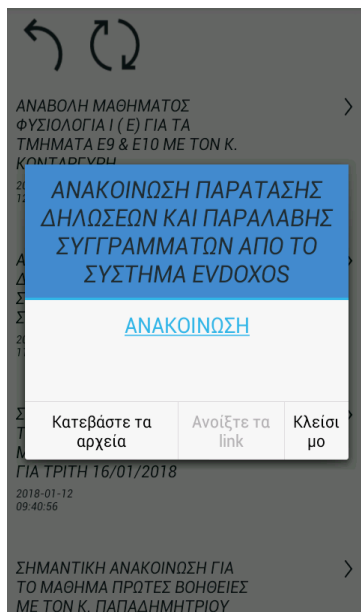
Εικόνα 27 Οθόνη ανακοινώσεων

Υπάρχουν ανακοινώσεις που περιέχουν μόνο κείμενο (εικόνα 28). Στην περίπτωση αυτή ο διάλογος που εμφανίζεται (alert dialog), τα κουμπιά που χρησιμοποιούνται για να κατεβούν αρχεία ή να ανοίξουν τα link, είναι απενεργοποιημένα. Επίσης μαζί με το κείμενο μπορεί να υπάρχουν και κάποια αρχεία, για να το κατεβάσει ο χρήστης ή κείμενο μαζί με κάποιο link για να ανοίξει ο χρήστης.



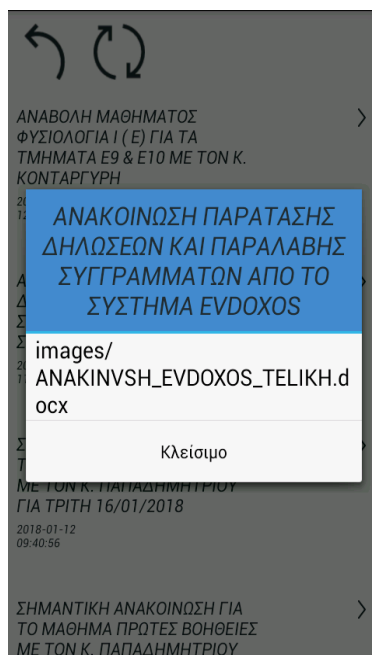
Εικόνα 28 Ανακοίνωση μόνο με κείμενο

Στην περίπτωση που περιέχει αρχεία ή link μαζί με το κείμενο της ανακοίνωσης, η μορφή είναι ίδια. Δηλαδή, εμφανίζεται ένας διάλογος που είναι απενεργοποιημένο μόνο το κουμπί για να ανοίξουν τα link ή να κατεβούν τα αρχεία αντίστοιχα (εικόνα 29).



Εικόνα 29 Ανακοίνωση με αρχεία (1)

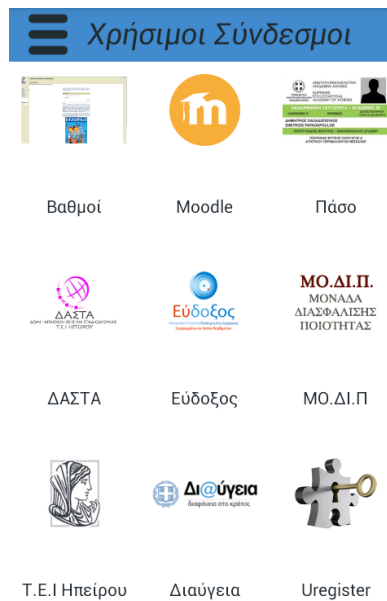
Τα δύο κουμπιά αυτά εμφανίζουν έναν ξεχωριστό διάλογο με μία λίστα, με τα αρχεία που είναι για κατέβασμα ή με τα link για άνοιγμα (εικόνα 30). Πατώντας πάνω σε μια από τις γραμμές τις λίστες, κατεβαίνει το αντίστοιχο αρχείο ή ανοίγει το αντίστοιχο link.



Εικόνα 30 Ανακοίνωση με αρχεία (2)

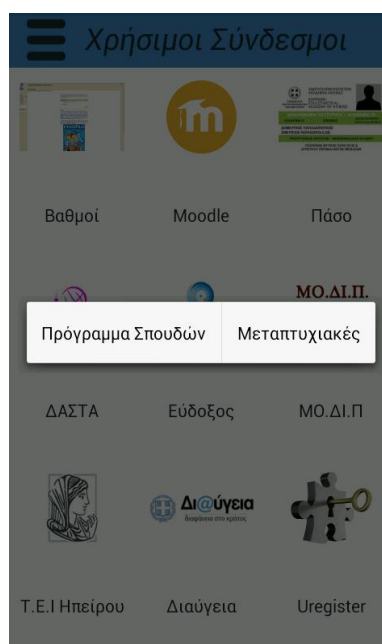
Η τελευταία μορφή ανακοινώσεων είναι να υπάρχουν και αρχεία και link. Σε αυτή τη περίπτωση και τα δυο κουμπιά είναι ενεργοποιημένα.

Η επόμενη επιλογή του μενού είναι οι «Χρήσιμοι Σύνδεσμοι» (εικόνα 31), που περιέχουν κάποια εικονίδια ιστοτόπων, που με απλό πάτημα πάνω τους, μεταφέρουν τον χρήστη στην αντίστοιχη ιστοσελίδα.



Εικόνα 31 Χρήσιμοι Σύνδεσμοι

Στη συνέχεια η επιλογή «Σπουδές» εμφανίζει ένα διάλογο με δύο κουμπιά (εικόνα 32), το «Πρόγραμμα Σπουδών» και το «Μεταπτυχιακές».



Εικόνα 32 Σπουδές

Το πρώτο κουμπί εμφανίζει μία οθόνη, (εικόνα 33), με ένα κουμπί που αν το πατήσει ο χρήστης κατεβάζει το αναλυτικό πρόγραμμα σπουδών, και την προεπισκόπηση των μαθημάτων της σχολής ανά εξάμηνο.



Λήψη Προγράμματος Σπουδών

Α' εξάμηνο				
ΜΑΘΗΜΑΤΑ	ΘΕΩΡΙΑ	ΕΡΓΑΣΤΗΡΙΑ	ΑΣΚΗΣΗ ΠΡΑΞΗΣ	ΩΡΕΣ ΕΡΓΑΣΙΑΣ
Ανατομία Ι	3	2	-	5
Φυσιολογία Ι	3	2	-	5
Εισαγωγή στην Νοσηλευτική	2	5	1	8
Βιολογία	3	-	1	4
Ψυχολογία της Υγείας	2	-	-	2
ΣΥΝΟΛΟ	13	9	2	24

Β' εξάμηνο				
ΜΑΘΗΜΑΤΑ	ΘΕΩΡΙΑ	ΕΡΓΑΣΤΗΡΙΑ	ΑΣΚΗΣΗ ΠΡΑΞΗΣ	ΩΡΕΣ ΕΡΓΑΣΙΑΣ
Ανατομία ΙΙ	2	2	-	4

Εικόνα 33 Πρόγραμμα σπουδών

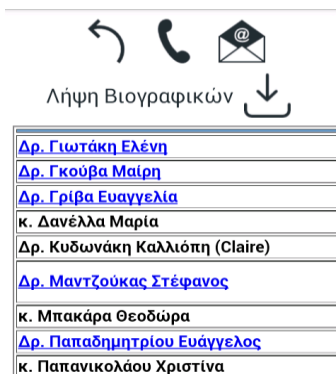
Το δεύτερο κουμπί πηγαίνει το χρήστη μέσω του browser στην ιστοσελίδα του μεταπτυχιακού της σχολής Νοσηλευτικής.

Η επιλογή «Ανθρώπινο Δυναμικό», εμφανίζει μία λίστα με τις κατηγορίες εργαζομένων στο τμήμα Νοσηλευτικής του Τ.Ε.Ι ΗΠΕΙΡΟΥ (εικόνα 34).

Ανθρώπινο Δυναμικό	
Μέλη ΔΕΠ	>
Εκτακτο Ε.Π	>
Ακαδημαϊκοί Υπότροφοι	>
ΕΤΕΠ	>
Διοικητικό Προσωπικό	>

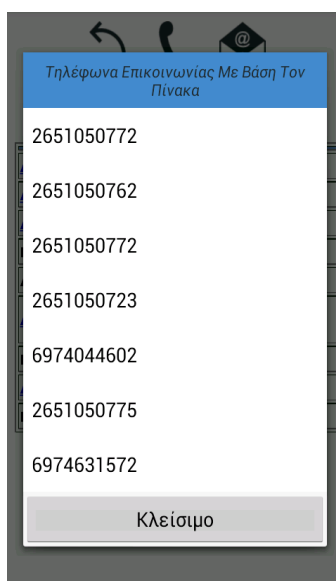
Εικόνα 34 Ανθρώπινο δυναμικό

Όλες οι επιλογές, εκτός από την «Έκτακτο Ε.Π.», που είναι link σε ένα αρχείο excel εκτός του site της Νοσηλευτικής, εμφανίζουν μία οθόνη με έναν πίνακα με το προσωπικό της κάθε κατηγορίας με τα στοιχεία επικοινωνίας τους (email,τηλέφωνα).



Εικόνα 35 Σελίδα ανθρώπινου δυναμικού

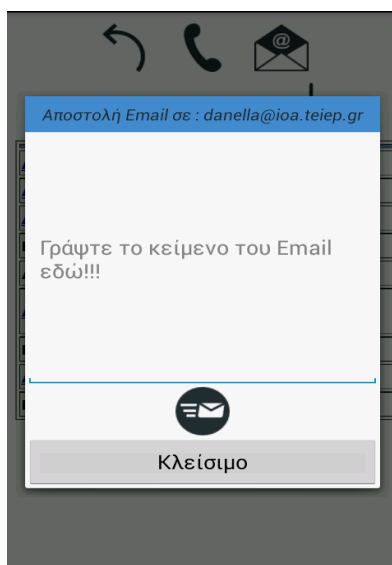
Ο χρήστης μπορεί να πάρει τηλέφωνο μέσω της εφαρμογής πατώντας σε αυτή την οθόνη την εικόνα με το τηλέφωνο (εικόνα 36). Εμφανίζει μία λίστα τηλεφώνων μέσω ενός διαλόγου με την σειρά του πίνακα. Η επιλογή οποιασδήποτε σειράς της λίστας πηγαίνει τον χρήστη κατευθείαν στην εφαρμογή για τηλέφωνα του smartphone του και ξεκινάει η κλήση.



Εικόνα 36 Τηλέφωνα επικοινωνίας

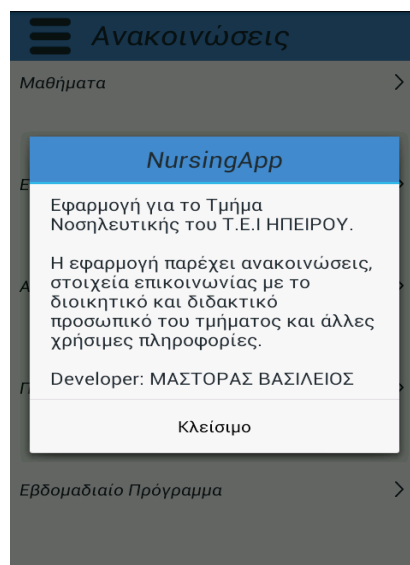
Για να αποσταλεί email μέσω της εφαρμογής πρέπει να πατηθεί το εικονίδιο με το φακελάκι. Έτσι εμφανίζεται ένας διάλογος ,με μία λίστα με email με την σειρά του πίνακα, όπως και με τα τηλέφωνα.

Αν πατηθεί οποιαδήποτε, εμφανίζει πάλι ένα ξεχωριστό διάλογο (εικόνα 37), με ένα πεδίο κειμένου και ένα εικονίδιο από κάτω για την αποστολή του email, που με την σειρά του εμφανίζει μία επιλογή, με ποια εφαρμογή θέλει ο χρήστης να το στείλει, ανάλογα με αυτές που είναι εγκατεστημένες στο κινητό του για την διαχείριση ηλεκτρονικού ταχυδρομείου.



Εικόνα 37 Αποστολή email

Τέλος η τελευταία επιλογή του μενού είναι οι πληροφορίες, που εμφανίζει έναν διάλογο με το τι περιέχει η εφαρμογή (εικόνα 37).



Εικόνα 38 Πληροφορίες εφαρμογής

4.4 Μελλοντικές επεκτάσεις.

Παρακάτω αναφέρονται σε τυχόν επεκτάσεις της εφαρμογής, έτσι ώστε να βελτιωθεί ή να αποκτήσει καινούριες λειτουργίες.

1. Προσθήκη επιλογής στο κεντρικό μενού της εφαρμογής, προβολής του Τ.Ε.Ι Ηπείρου Σ.Ε.Υ.Π Ιωαννίνων στον χάρτη (Google Maps).
2. Βελτίωση γραφικής διεπαφής της εφαρμογής, αντικαθιστώντας τους κώδικες γραφικών με χρήση της γλώσσας Java, με αρχεία κώδικα Xml.
3. Βελτίωση διαχείρισης ανακοινώσεων.

5 Παράρτημα (ο κώδικας)

Ο κώδικας της κλάσης **MainActivity**, που δημιουργεί όλο το κεντρικό μενού της εφαρμογής:

```
package com.example.ptixiaki;

import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.R.color;
import android.annotation.SuppressLint;
import android.app.ActionBar;
import android.app.ActionBar.LayoutParams;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DownloadManager;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
```

```

import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Point;
import android.graphics.Typeface;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v4.widget.DrawerLayout;
import android.text.Html;
import android.text.InputType;
import android.text.Spannable;
import android.text.SpannableString;
import android.text.method.LinkMovementMethod;
import android.text.style.ForegroundColorSpan;
import android.util.DisplayMetrics;
import android.util.Log;
import android.view.Display;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.view.animation.Animation;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.AbsListView;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.RelativeLayout;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewFlipper;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.TabHost.TabSpec;

public class MainActivity extends Activity {
    private String TAG = MainActivity.class.getSimpleName();
    DrawerLayout mainLayout=null;
    LinearLayout ll=null;
    static TextView t=null;
    ArrayList<ListItem> items=null;
    ArrayList<Anakoinoseis>analist=null;

```

```

        ArrayList<Prosopiko>proslis=null;
        ListView list=null,lv=null,prosl=null;
        ListAdapter adapter=null;
        AnaAdapter adapter1=null;
        ProsAdapter adapter2=null;
        int width=0,height=0,x=0;;
        Bitmap icon;
        ViewFlipper flipper;
        Context ctx;
        String type=null,link=null,link1=null;
        WebView wv=null;
        String jsonStr1,jsonStr2;
        private static String url1="http://nursing.ioa.teiep.gr/nursing-
app/ana_cat.php?call=all";
        private static String url2="http://nursing.ioa.teiep.gr/nursing-
app/prosopiko.php?call=all";

        private boolean isNetworkAvailable() {
            ConnectivityManager connectivityManager
                = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
            NetworkInfo activeNetworkInfo =
connectivityManager.getActiveNetworkInfo();
            return activeNetworkInfo != null &&
activeNetworkInfo.isConnected();
        }

        public void makeHeader(String title)
        {
            LinearLayout header=new LinearLayout(this);
            ll.addView(header);
            header.setBackgroundColor(Color.rgb(66,139,202));
            LinearLayout.LayoutParams
hparams=(LinearLayout.LayoutParams)header.getLayoutParams();
            hparams.width=width;
            hparams.height=height/12;
            header.setLayoutParams(hparams);
            ImageView openDrawer=new ImageView(this);
            header.addView(openDrawer);
            t=new TextView(this);
            t.setTextColor(Color.rgb(37,45,53));
            header.addView(t);
            LinearLayout.LayoutParams
oparams=(LinearLayout.LayoutParams)openDrawer.getLayoutParams();
            oparams.leftMargin=width/20;
            oparams.topMargin=hparams.height/10;
            oparams.width=width/10;
            oparams.height=hparams.height-2*oparams.topMargin;
            openDrawer.setLayoutParams(oparams);
            Bitmap b1=BitmapFactory.decodeResource(getResources(),
R.drawable.menu2);
            b1=b1.createScaledBitmap(b1, oparams.width, oparams.height,
true);

            openDrawer.setImageBitmap(b1);
            openDrawer.setClickable(true);
            openDrawer.setOnClickListener(new OnClickListener()
            {
                @Override
                public void onClick(View v) {
                    if(mainLayout.isDrawerOpen(list))
mainLayout.closeDrawer(list);

```

```

        else mainLayout.openDrawer(list);
    }
});
LinearLayout.LayoutParams
tparams=(LinearLayout.LayoutParams)t.getLayoutParams();
tparams.leftMargin=width/20;
tparams.topMargin=hparams.height/10;
t.setLayoutParams(tparams);
t.setTypeface(null, Typeface.ITALIC);
t.setTextSize(24);
t.setText(title);
}

public void makeHome()
{
    l1=new LinearLayout(this);
    mainLayout.addView(l1);
    l1.setId(100);
    l1.setOrientation(LinearLayout.VERTICAL);
    DrawerLayout.LayoutParams
l1params=(DrawerLayout.LayoutParams)l1.getLayoutParams();
l1params.height=height-height/40;
l1params.width=width;
l1.setLayoutParams(l1params);
l1.setBackgroundColor(Color.WHITE);
}

public void arxikh()
{
    LinearLayout l2=new LinearLayout(this);
    l2.setGravity(Gravity.CENTER);
    l2.setOrientation(LinearLayout.VERTICAL);
    l2.setBackgroundColor(Color.rgb(37,45,53));
    flipper.addView(l2);

    ImageView im1=new ImageView(this);
    l2.addView(im1);
    Bitmap
bm1=BitmapFactory.decodeResource(getResources(),R.drawable.logoapp);
    final int width=89;
    final int height=99;
    bm1=bm1.createScaledBitmap(bm1, width,height, true);
    im1.setImageBitmap(bm1);

    TextView home=new TextView(this);
    l2.addView(home);
    home.setText("Καλώς ήλθατε στην εφαρμογή του τμήματος
    νοσηλευτικής του Τ.Ε.Ι Ηπείρου");
    home.setGravity(Gravity.CENTER);
    home.setTextColor(Color.WHITE);
    home.setTypeface(null, Typeface.ITALIC);
    home.setShadowLayer(2, 2, 2, Color.BLACK);
    home.setTextSize(25);
}

public void anakoinoseis()
{
    analist=new ArrayList<Anakoinoseis>();
    lv=new ListView(this);
    final int dividerHeight=80;
    LinearLayout l3=new LinearLayout(this);

```

```

l3.addView(lv);
flipper.addView(l3);

final HttpHandler gb = new HttpHandler();
jsonStr1 = gb.makeServiceCall(url1);
Log.e(TAG, "Response from url: " + jsonStr1);
if (jsonStr1!=null)
{
    try {
        JSONObject jsonObj=new JSONObject(jsonStr1);
        JSONArray
anakoinoseis=jsonObj.getJSONArray("anakoinoseis");
        for (int i=0;i<anakoinoseis.length();i++)
        {
            JSONObject a=anakoinoseis.getJSONObject(i);
            String id=a.getString("id");
            String title=a.getString("title");
            Bitmap
bm=BitmapFactory.decodeResource(getResources(),R.drawable.next);
            analist.add(new Anakoinoseis(title,bm));
        }
    } catch (final JSONException e) {
        Log.e(TAG, "Json parsing error: " + e.getMessage());
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
else
{
    Log.e(TAG, "Couldn't get json from server.");
}

adapter1=new
AnaAdapter(this,android.R.layout.simple_list_item_1,analist);
LinearLayout.LayoutParams
lvparams=(LinearLayout.LayoutParams)lv.getLayoutParams();
lvparams.height=DrawerLayout.LayoutParams.MATCH_PARENT;
lv.setLayoutParams(lvparams);
lv.setAdapter(adapter1);
lv.setBackgroundColor(Color.WHITE);
lv.setFastScrollEnabled(true);

lv.setOverScrollMode(AbsListView.OVER_SCROLL_IF_CONTENT_SCROLLS);
lv.setDivider(null);
lv.setDividerHeight(dividerHeight);
lv.setSelector(R.drawable.list_item_selector);
lv.setOnItemClickListener(new OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View
view,
                                int position, long id) {

        if(jsonStr1 != null)
        {
            try {
                JSONObject jsonObj = new
JSONObject(jsonStr1);
                JSONArray anakoinoseis =
jsonObj.getJSONArray("anakoinoseis");
                JSONObject json =
anakoinoseis.getJSONObject(position);

```

```

        String Id1 = json.getString("id");

        HttpHandler.catid=Integer.parseInt(Id1.toString());

        Intent I=new
Intent(MainActivity.this,AnaTitle.class);
        startActivity(I);
    } catch (final JSONException e) {
        Log.e(TAG, "Json parsing error: " +
e.getMessage());

        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    }
else
{
    Log.e(TAG, "Couldn't get json from server.");
}
});
}

public void xrisimoiSindesmroi()
{
    RelativeLayout rel=new RelativeLayout(this);
    flipper.addView(rel);
    final LayoutInflater factory=getLayoutInflater();
    final View
textEntryView=factory.inflate(R.layout.xrisimoi_sindesmroi,null);
    LinearLayout lin=(LinearLayout)
textEntryView.findViewById(R.id.lin);
    rel.addView(lin);

    ImageView im1=(ImageView) findViewById(R.id.vathmoi);
    im1.setOnClickListener(new OnClickListener()
    {
        @Override
        public void onClick(View v) {
            Intent browserIntent = new
Intent(Intent.ACTION_VIEW,
Uri.parse("https://gmweb.teiep.gr/unistudent/login.asp?mnuID=student"));
            startActivity(browserIntent);
        }
    });

    ImageView im2=(ImageView) findViewById(R.id.moodle);
    im2.setOnClickListener(new OnClickListener()
    {
        @Override
        public void onClick(View v) {
            Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://moodle.ioa.teiep.gr/"));
            startActivity(browserIntent);
        }
    });

    ImageView im3=(ImageView) findViewById(R.id.paso);
    im3.setOnClickListener(new OnClickListener()
    {
        @Override
        public void onClick(View v) {

```



```

        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("https://submit-
academicid.minedu.gov.gr/"));
        startActivity(browserIntent);
    }
});
ImageView im4=(ImageView) findViewById(R.id.dasta);
im4.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://dasta.teiep.gr/"));
        startActivity(browserIntent);
    }
});
ImageView im5=(ImageView) findViewById(R.id.eudoxos);
im5.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://eudoxus.gr/"));
        startActivity(browserIntent);
    }
});
ImageView im6=(ImageView) findViewById(R.id.modip);
im6.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://modip.teiep.gr/"));
        startActivity(browserIntent);
    }
});
ImageView im7=(ImageView) findViewById(R.id.tei);
im7.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://www.teiep.gr/"));
        startActivity(browserIntent);
    }
});
ImageView im8=(ImageView) findViewById(R.id.diavgeia);
im8.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW,
Uri.parse("https://et.diavgeia.gov.gr/f/teiep"));
        startActivity(browserIntent);
    }
});
ImageView im9=(ImageView) findViewById(R.id.Uregister);
im9.setOnClickListener(new OnClickListener()
{
    @Override

```

```

        public void onClick(View v) {
            Intent browserIntent = new
Intent(Intent.ACTION_VIEW,
Uri.parse("https://mypassword.teiep.gr/reset_password.php"));
            startActivity(browserIntent);
        }
    });

}

public void prosopiko()
{
    prosl=new ArrayList<Prosopiko>();
    prosl=new ListView(this);
    final int dividerHeight=80;
    LinearLayout l4=new LinearLayout(this);
    l4.addView(prosl);
    flipper.addView(l4);

    final HttpHandler gb = new HttpHandler();
    jsonStr2 = gb.makeServiceCall(url2);
    Log.e(TAG, "Response from url: " + jsonStr2);
    if(jsonStr2!=null)
    {
        try {
            JSONObject jsonObj =new JSONObject(jsonStr2);
            JSONArray prosopiko=jsonObj.getJSONArray("prosopiko");
            for (int i=0;i<prosopiko.length();i++)
            {
                JSONObject a=prosopiko.getJSONObject(i);
                String id=a.getString("id");
                String title=a.getString("title");
                String link=a.getString("link");
                String type=a.getString("type");
                Bitmap
bm=BitmapFactory.decodeResource(getResources(),R.drawable.next);
                prosl.add(new Prosopiko(title,bm,type,link));
            }
        } catch (final JSONException e) {
            Log.e(TAG, "Json parsing error: " + e.getMessage());
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    else
    {
        Log.e(TAG, "Couldn't get json from server.");
    }
    adapter2=new
ProsAdapter(this,android.R.layout.simple_list_item_1,prosl);
    LinearLayout.LayoutParams
lvparams=(LinearLayout.LayoutParams)lv.getLayoutParams();
    lvparams.height=DrawerLayout.LayoutParams.MATCH_PARENT;
    prosl.setLayoutParams(lvparams);
    prosl.setAdapter(adapter2);
    prosl.setBackgroundColor(Color.WHITE);
    prosl.setFastScrollEnabled(true);

    prosl.setOverScrollMode(AbsListView.OVER_SCROLL_IF_CONTENT_SCROLLS)
;
    prosl.setDivider(null);

```

```

        prosl.setDividerHeight(dividerHeight);
        prosl.setSelector(R.drawable.list_item_selector);
        prosl.setOnItemClickListener(new OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View
view,
                                int position, long id) {
                String test=prosl.get(position).link;
                String s=test.substring(0,5);
                if(s.equals("index"))
                {

                    HttpHandler.plinkid=prosl.get(position).link;
                    Intent I=new
Intent(MainActivity.this,ProsIntro.class);
                    startActivity(I);

                }
                else if(!s.equals("index"))
                {
                    Intent browserIntent=new
Intent(Intent.ACTION_VIEW, Uri.parse(prosl.get(position).link));
                    startActivity(browserIntent);

                }

            }

        });

    }

    public void makeList()
    {
        items=new ArrayList<ListItem>();
        list=new ListView(this);

        ImageView icon=new ImageView(this);
        Bitmap
bm0=BitmapFactory.decodeResource(getResources(),R.drawable.logo_gr);
        double ratio=bm0.getHeight()*10.0/bm0.getWidth();
        final int desiredWidth=30;
        final int desiredHight=(int) (ratio*desiredWidth);
        bm0=bm0.createScaledBitmap(bm0,desiredWidth+230
,desiredHight-10 , true);
        list.addHeaderView(icon);
        icon.setBackgroundColor(Color.rgb(37,45,53));
        icon.setImageBitmap(bm0);
        mainLayout.addView(list);
        DrawerLayout.LayoutParams
cparams=(DrawerLayout.LayoutParams)list.getLayoutParams();
        cparams.width=3*width/4;
        cparams.height=DrawerLayout.LayoutParams.MATCH_PARENT;
        cparams.gravity=Gravity.START;
        list.setLayoutParams(cparams);

        Bitmap
bm=BitmapFactory.decodeResource(getResources(),R.drawable.home);
        bm=bm.createScaledBitmap(bm, width, height, true);
        items.add(new ListItem("Αρχική",bm));

        Bitmap
bm1=BitmapFactory.decodeResource(getResources(),R.drawable.anakinoseis1);
        bm1=bm1.createScaledBitmap(bm1, width, height, true);

```

```

        items.add(new ListItem("Ανακοινώσεις",bm1));

        Bitmap
bm2=BitmapFactory.decodeResource(getResources(),R.drawable.link);
bm2=bm2.createScaledBitmap(bm2, width, height, true);
        items.add(new ListItem("Χρήσιμοι Σύνδεσμοι",bm2));

        Bitmap
bm3=BitmapFactory.decodeResource(getResources(),R.drawable.akadimaika);
bm3=bm3.createScaledBitmap(bm3, width, height, true);
        items.add(new ListItem("Σπουδές",bm3));

        Bitmap
bm5=BitmapFactory.decodeResource(getResources(),R.drawable.prosopiko);
bm5=bm5.createScaledBitmap(bm5, width, height, true);
        items.add(new ListItem("Ανθρώπινο Δυναμικό",bm5));

        Bitmap
bm6=BitmapFactory.decodeResource(getResources(),R.drawable.coding);
bm6=bm6.createScaledBitmap(bm6, width, height, true);
        items.add(new ListItem("Πληροφορίες",bm6));

        adapter=new
ListAdapter(this,android.R.layout.simple_list_item_1,items);
        list.setAdapter(adapter);
        list.setBackgroundColor(Color.WHITE);
        list.setDivider(null);
        list.setSelector(R.drawable.list_item_selector);
        list.setOnItemClickListener(new OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View
view,

                                int position, long id) {

                if(position==0) return ;
                if(position==1)
                {
                    flipper.setDisplayedChild(0);
                    MainActivity.t.setText("Αρχική");
                }
                else
                if(position==2)
                {
                    flipper.setDisplayedChild(1);
                    MainActivity.t.setText("Ανακοινώσεις");
                }
                else
                if(position==3)
                {
                    flipper.setDisplayedChild(2);
                    MainActivity.t.setText("Χρήσιμοι
Σύνδεσμοι");
                }
                else
                if(position==4)
                {
                    AlertDialog.Builder alertDialog=new
AlertDialog.Builder(MainActivity.this);
                    alertDialog.setNegativeButton("Πρόγραμμα
Σπουδών", new DialogInterface.OnClickListener() {

```

```

@Override
public void onClick(DialogInterface
dialog, int which) {
    Intent I=new
Intent(MainActivity.this,Programma.class);
    startActivity(I);
}
});
alertDialog.setPositiveButton("Μεταπτυχιακές",
new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface
dialog, int which) {
        Intent browserIntent = new
Intent(Intent.ACTION_VIEW, Uri.parse("http://nursing-
msc.ioa.teiep.gr/"));
        startActivity(browserIntent);
    }
});
alertDialog.show();
}
else
if(position==5)
{
    flipper.setDisplayedChild(3);
    MainActivity.t.setText("Ανθρώπινο
Δυναμικό");
}
else
if(position==6)
{
    AlertDialog.Builder alertDialog=new
AlertDialog.Builder(MainActivity.this);
    TextView title = new TextView(MainActivity.this);
    title.setText("NursingApp");
    title.setBackgroundColor(Color.rgb(66,139,202));
    title.setPadding(10,10,10,10);
    title.setGravity(Gravity.CENTER);
    title.setTypeface(null, Typeface.ITALIC);
    title.setTextColor(Color.rgb(37,45,53));
    title.setTextSize(20);
    alertDialog.setMessage("Εφαρμογή για το Τμήμα
Νοσηλευτικής του Τ.Ε.Ι ΗΠΕΙΡΟΥ. \n"
        +" \n"
        +"Η εφαρμογή παρέχει ανακοινώσεις, στοιχεία
επικοινωνίας με το διοικητικό και διδακτικό προσωπικό του τμήματος και
άλλες χρήσιμες πληροφορίες. \n"
        +" \n"
        +"Developer: ΜΑΣΤΟΡΑΣ ΒΑΣΙΛΕΙΟΣ");
    alertDialog.setPositiveButton("Κλείσιμο",
new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface
dialog, int which) {
            dialog.cancel();
        }
    });
    alertDialog.setCustomTitle(title);
}
}

```

```

        AlertDialog dialog=alertDialog.show();
        TextView
messageText=(TextView)dialog.findViewById(android.R.id.message);

        messageText.setTextColor(Color.rgb(37,45,53));
        messageText.setGravity(Gravity.LEFT);
        messageText.setTextSize(15);
    }
    mainLayout.closeDrawer(list);
}
});
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    mainLayout=new DrawerLayout(this);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setTheme(R.style.AppTheme);
    setContentView(mainLayout);
    Display display = getWindowManager().getDefaultDisplay();
    width = display.getWidth();
    height = display.getHeight();
    flipper=new ViewFlipper(this);
    makeHome();
    makeHeader("Αρχική");
    ll.addView(flipper);
    LinearLayout.LayoutParams
flparams=(LinearLayout.LayoutParams) flipper.getLayoutParams();
    flparams.height=LinearLayout.LayoutParams.MATCH_PARENT;
    flparams.width=LinearLayout.LayoutParams.MATCH_PARENT;
    flipper.setLayoutParams(flparams);
    arxikh();
    anakoinoseis();
    xrisimoiSindesmoi();
    prosopiko();
    makeList();
    if(!isNetworkAvailable())
    {
        AlertDialog.Builder alertDialog=new
AlertDialog.Builder(MainActivity.this);
        TextView title = new TextView(MainActivity.this);
        title.setText("Χωρίς Σύνδεση");
        title.setBackgroundColor(Color.WHITE);
        title.setPadding(10,10,10,10);
        title.setGravity(Gravity.CENTER);
        title.setTypeface(null, Typeface.ITALIC);
        title.setTextColor(Color.rgb(37,45,53));
        title.setTextSize(20);
        alertDialog.setMessage("Συνδεθείτε στα δεδομένα ή στο WiFi
και μετά πατήστε το κουμπί "Ανανέωση");
        alertDialog.setNegativeButton("Ανανέωση", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int
which) {

                Intent intent = getIntent();

```

```

        finish();
        startActivity(intent);
    }
});
alertDialog.setCustomTitle(title);
AlertDialog dialog=alertDialog.show();
TextView
messageText=(TextView)dialog.findViewById(android.R.id.message);
messageText.setTextColor(Color.rgb(37,45,53));
messageText.setGravity(Gravity.CENTER);
    }
}
}

```

Ο κώδικας των κλάσεων **ListItem** και **ListAdapter** που ακολουθούν, δημιουργούν την μορφή του κεντρικού μενού της εφαρμογής. Όλες οι λίστες στην εφαρμογής που περιέχουν πέρα από απλό κείμενο και εικόνες, έχουν δημιουργηθεί με τον ίδιο τρόπο. Για αυτό τον λόγο δεν παρουσιάζονται.

ListItem:

```

package com.example.ptixiaki;

import android.graphics.Bitmap;

public class ListItem {
    public String text;
    public Bitmap icon;
    public ListItem(String txt, Bitmap i)
    {
        text=txt;
        icon=i;
    }
}

```

ListAdapter:

```

package com.example.ptixiaki;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Typeface;
import android.text.Html;
import android.util.Log;
import android.view.Display;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;

```

```

import android.widget.TextView;

public class ListAdapter extends ArrayAdapter{
    ArrayList<ListItem> items=new ArrayList<ListItem>();
    Context mContext=null;
    public ListAdapter(Context context,int resource,ArrayList<ListItem>
t)
    {
        super(context, resource,t);
        mContext=context;
        items=t;
    }

    public View getView(int index, View convertView, ViewGroup
viewGroup)
    {
        Display display = ((Activity)
mContext).getWindowManager().getDefaultDisplay();
        int width = display.getWidth();
        int height = display.getHeight();
        LinearLayout l=new LinearLayout(mContext);

        ImageView iml=new ImageView(mContext);
        l.addView(iml);
        LinearLayout.LayoutParams
imlparams=(LinearLayout.LayoutParams)iml.getLayoutParams();
        imlparams.width=width/9;
        imlparams.height=height/14;
        iml.setLayoutParams(imlparams);
        Bitmap bml=items.get(index).icon;
        bml=bml.createScaledBitmap(bml, imlparams.width,
imlparams.height, true);
        iml.setImageBitmap(bml);
        iml.setPadding(12,12,12,12);

        TextView t=new TextView(mContext);
        l.addView(t);
        LinearLayout.LayoutParams
tparams=(LinearLayout.LayoutParams)t.getLayoutParams();
        tparams.height=imlparams.height;
        tparams.width=7 *width/10;
        tparams.leftMargin=width/20;
        t.setLayoutParams(tparams);
        t.setText(items.get(index).text);
        t.setTypeface(null, Typeface.ITALIC);
        t.setTextColor(Color.rgb(37,45,53));
        t.setTextSize(15);
        t.setGravity(Gravity.FILL);

        return l;
    }
}

```

Ακολουθεί το αρχείο Xml **xrisimoi_sinedsmoi** που κάνει την μορφοποίηση των
χρήσιμων συνδέσμων.

```

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

```



```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/lin"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="com.example.xrisimoisindesoi.MainActivity"
tools:layout_editor_absoluteX="8dp"
tools:layout_editor_absoluteY="8dp">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2">

                    <ImageView
                        android:id="@+id/vathmoi"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:layout_alignParentBottom="true"
                        android:padding="10dp"
android:src="@drawable/vathmoi"/>

                    </RelativeLayout>

                    <RelativeLayout
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:layout_weight="0.2">

                            <TextView
                                android:id="@+id/textView"
                                android:layout_width="match_parent"
                                android:layout_height="match_parent"
                                android:layout_centerHorizontal="true"
                                android:layout_centerVertical="true"
                                android:gravity="center"
                                android:text="Βαθμοί "
                                android:textColor="@color/splashcolor" />

                            </RelativeLayout>
                        </LinearLayout>

```

```

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <ImageView
                android:id="@+id/moodle"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:padding="10dp"
                android:src="@drawable/moodle"/>

            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.2">

                <TextView
                    android:id="@+id/textView5"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:gravity="center"
                    android:text="Moodle"
                    android:textColor="@color/splashcolor" />

                </RelativeLayout>
            </LinearLayout>
        </RelativeLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2">

```

```

        <ImageView
            android:id="@+id/paso"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:padding="10dp"
            android:src="@drawable/paso"/>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="0.2">

        <TextView
            android:id="@+id/textView10"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:gravity="center"
            android:text="Πάσο"
            android:textColor="@color/splashcolor" />
    </RelativeLayout>
</LinearLayout>
</RelativeLayout>
</LinearLayout>
</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2">

                    <ImageView
                        android:id="@+id/dasta"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"

```

```

        android:padding="10dp"
        android:src="@drawable/dasta"/>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="0.2">

        <TextView
            android:id="@+id/textView11"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:gravity="center"
            android:text="ΔΔΕΤΑ"
            android:textColor="@color/splashcolor" />
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <ImageView
                android:id="@+id/eudoxos"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="10dp"
                android:src="@drawable/logo"/>
            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.2">

                <TextView
                    android:id="@+id/textView12"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:gravity="center"
                    android:text="Εύδοξος"
                    android:textColor="@color/splashcolor" />
                </RelativeLayout>
            </LinearLayout>
        </RelativeLayout>
    </LinearLayout>

```

```

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <ImageView
                android:id="@+id/modip"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="10dp"
                android:src="@drawable/modip"/>
            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.2">

                <TextView
                    android:id="@+id/textView13"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:gravity="center"
                    android:text="MO.ΔΙ.Π"
                    android:textColor="@color/splashcolor" />
                </RelativeLayout>
            </LinearLayout>
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <LinearLayout

```

```

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <ImageView
                android:id="@+id/tei"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="10dp"
                android:src="@drawable/tei"/>
            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.2">

                <TextView
                    android:id="@+id/textView14"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:gravity="center"
                    android:text="Τ.Ε.Ι Ημερίδα"
                    android:textColor="@color/splashcolor" />
                </RelativeLayout>
            </LinearLayout>
        </RelativeLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

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

                <RelativeLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_weight="0.2">

                    <ImageView
                        android:id="@+id/diavgeia"
                        android:layout_width="match_parent"
                        android:layout_height="match_parent"
                        android:layout_alignParentLeft="true"
                        android:layout_alignParentStart="true"
                        android:layout_alignParentTop="true"
                        android:padding="10dp"
                        android:src="@drawable/diavgeia"/>
                    </RelativeLayout>

```

```

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <TextView
                android:id="@+id/textView15"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_centerHorizontal="true"
                android:layout_centerVertical="true"
                android:gravity="center"
                android:text="Διαύγεια"
                android:textColor="@color/splashcolor" />

        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.2">

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

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.2">

            <ImageView
                android:id="@+id/Uregister"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:padding="10dp"
                android:src="@drawable/fopass"/>
            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="0.2">

                <TextView
                    android:id="@+id/textView16"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:gravity="center"
                    android:text="Uregister"
                    android:textColor="@color/splashcolor" />
                </RelativeLayout>
            </LinearLayout>
        </RelativeLayout>
    </LinearLayout>
</RelativeLayout>
</LinearLayout>

```

Η κλάση **Programma** δημιουργεί την οθόνη που εμφανίζει το πρόγραμμα σπουδών.

```
package com.example.ptixiaki;

import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.app.ActionBar.LayoutParams;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.DownloadManager;
import android.app.ProgressDialog;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.StrictMode;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

public class Programma extends Activity {
    private String TAG = Programma.class.getSimpleName();
    WebView wv=null;
    public static String introtext=null;
    LinearLayout mainLayout=null;
    private String url =
"http://83.212.97.229/nursingmet/webservices/article.php?call=all";
    private String url2 ="http://83.212.97.229/nursingmet/";
    private ProgressDialog progressDialog;

    private void startLoading(String message)
    {
        progressDialog = new ProgressDialog(Programma.this);
        progressDialog.setMessage(message);
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }
}
```



```

    }

    private void stopLoading()
    {
        proDialog.dismiss();
        proDialog = null;
    }

    String downloadLinks(String text)
    {
        String regex = "((http:\\\\|https:\\\\|)?(www.)?((\\|)*[a-zA-Z0-9-]*|[a-zA-Z0-9-]){2,}\\|\\.){1,4}([a-zA-Z]){2,6}(\\|/([a-zA-Z-_|\\\\|\\.0-9#?:=&;,]*)?)?)";
        Pattern p = Pattern.compile(regex);
        Matcher m = p.matcher(text);
        String apot=null;

        while(m.find())
        {
            String urlStr = m.group();
            char[] stringArray1 = urlStr.toCharArray();
            if(urlStr.startsWith("(") && urlStr.endsWith(")"))
            {
                char[] stringArray=urlStr.toCharArray();
                char[] newArray=new char[stringArray.length-2];

                System.arraycopy(stringArray,1,newArray,0,stringArray.length-2);
                urlStr=new String(newArray);
                System.out.println("Finally Url =" +newArray.toString());
            }
            System.out.println("...Url..." +urlStr);
            apot=urlStr;
        }
        return apot;
    }

    String retrieveLinks(String text)
    {
        String regex = "((http:\\\\|https:\\\\|)?(www.)?(([a-zA-Z0-9-]*|[a-zA-Z0-9-]){2,}\\|\\.){1,4}([a-zA-Z]){2,6}(\\|/([a-zA-Z-_|\\\\|\\.0-9#?:=&;,]*)?)?)";
        Pattern p = Pattern.compile(regex);
        Matcher m = p.matcher(text);
        String apot=null;
        while(m.find())
        {
            String urlStr = m.group();
            char[] stringArray1 = urlStr.toCharArray();
            if(urlStr.startsWith("(") && urlStr.endsWith(")"))
            {
                char[] stringArray=urlStr.toCharArray();
                char[] newArray=new char[stringArray.length-2];

                System.arraycopy(stringArray,1,newArray,0,stringArray.length-2);
                urlStr=new String(newArray);
                System.out.println("Finally Url =" +newArray.toString());
            }
            System.out.println("...Url..." +urlStr);
            apot=urlStr;
        }
        return apot;
    }

```

```

    }

    void programma ()
    {
        final HttpHandler gb = new HttpHandler();
        final String jsonStr = gb.makeServiceCall(url);
        Log.e(TAG, "Response from url: " + jsonStr);
        if (jsonStr!=null)
        {
            try {
                JSONObject jsonObj=new JSONObject(jsonStr);
                JSONArray article=jsonObj.getJSONArray("article");
                for (int i=0; i<article.length();i++)
                {
                    JSONObject a=article.getJSONObject(i);
                    introtext=a.getString("introtext");
                }
            } catch (final JSONException e) {
                Log.e(TAG, "Json parsing error: " + e.getMessage());
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        else
        {
            Log.e(TAG, "Couldn't get json from server.");
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    Toast.makeText(getApplicationContext(),
                        "Συνδεθείτε στα δεδομένα ή στο WiFi",
                        Toast.LENGTH_LONG)
                        .show();
                }
            });
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        StrictMode.ThreadPolicy policy=new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        mainLayout=new LinearLayout(this);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setTheme(R.style.AppTheme);
        setContentView(mainLayout);
        mainLayout.setOrientation(LinearLayout.VERTICAL);
        final ImageView iml=new ImageView(this);
        Bitmap
        bml=BitmapFactory.decodeResource(getResources(),R.drawable.back1);
        final int width=66;
        final int height=66;
        bml=bml.createScaledBitmap(bml, width,height, true);
        iml.setImageBitmap(bml);
        mainLayout.addView(iml);
        iml.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v) {

```

```

        onBackPressed();
    }
});
im1.setPadding(15,15,15,15);
im1.setScaleType(ImageView.ScaleType.FIT_START);
Button b=new Button(this);
mainLayout.addView(b);
b.setText("Λήψη Προγράμματος Σπουδών");
b.setTypeface(null, Typeface.ITALIC);
b.setTextColor(Color.rgb(37,45,53));
b.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        startLoading("Λήψη...");
        new Thread(new Runnable() {
            @Override
            public void run()
            {
                String url =
url2+"/"+downloadLinks(introtext);
                DownloadManager.Request request = new
DownloadManager.Request(Uri.parse(url));
                request.setDescription("NursingApp");

                request.setTitle(retrieveLinks(introtext).toString());

                if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.HONEYCOMB)
                {

                    request.allowScanningByMediaScanner();

                    request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISI
BLE_NOTIFY_COMPLETED);

                }
                DownloadManager manager =
(DownloadManager) getSystemService(Context.DOWNLOAD_SERVICE);
                manager.enqueue(request);
                runOnUiThread(new Runnable() {
                    @Override
                    public void run()
                    {
                        stopLoading();
                    }
                });
            }
        }).start();
    }
});
programma();
wv=new WebView(this);
wv.getSettings().setJavaScriptEnabled(true);

wv.loadData(introtext.substring(introtext.indexOf('\n'),introtext.length(
)-1),"text/html; charset=UTF-8", null);
mainLayout.addView(wv);
LinearLayout.LayoutParams
wparams=(LinearLayout.LayoutParams)wv.getLayoutParams();
wparams.height=LayoutParams.MATCH_PARENT;
wparams.width=LayoutParams.MATCH_PARENT;
wv.setLayoutParams(wparams);

```

```

    }

}

```

Η κλάση **ProsIntro**, δημιουργεί την οθόνη με τα στοιχεία επικοινωνίας του προσωπικού της σχολής.

```

package com.example.ptixiaki;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Iterator;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DownloadManager;
import android.app.ProgressDialog;
import android.app.ActionBar.LayoutParams;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.Rect;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v4.widget.DrawerLayout;
import android.telephony.PhoneNumberUtils;
import android.text.Html;
import android.text.InputType;
import android.text.method.LinkMovementMethod;
import android.util.Log;
import android.view.Display;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;
import android.view.Window;
import android.view.View.OnClickListener;
import android.view.WindowManager;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.AbsListView;

```

```

import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class ProsIntro extends Activity{
    WebView wv=null;
    LinearLayout mainLayout=null;
    private String TAG=ProsIntro.class.getSimpleName();
    public static String introtext=null;
    private ProgressDialog progressDialog;
    private String url="http://nursing.ioa.teiep.gr/nursing-
app/prosopiko_intr.php?call=one&plinkid="+HttpHandler.plinkid.substring(H
ttpHandler.plinkid.length()-2);
    private String url2 ="http://nursing.ioa.teiep.gr/";
    ArrayList<String> s1=null;
    LinearLayout ld;

    private void startLoading(String message)
    {
        progressDialog=new ProgressDialog(ProsIntro.this);
        progressDialog.setMessage(message);
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

    private void stopLoading()
    {
        progressDialog.dismiss();
        progressDialog=null;
    }

    void prosintro()
    {
        final HttpHandler gb = new HttpHandler();
        final String jsonStr = gb.makeServiceCall(url);
        Log.e(TAG, "Response from url: " + jsonStr);
        if (jsonStr!=null)
        {
            try {
                JSONObject jsonObj=new JSONObject(jsonStr);
                JSONArray prosIntro=jsonObj.getJSONArray("prosintr");
                for (int i=0;i<prosIntro.length();i++)
                {
                    JSONObject a=prosIntro.getJSONObject(i);
                    introtext=a.getString("introtext");
                }
            } catch (final JSONException e) {
                Log.e(TAG, "Json parsing error: " + e.getMessage());
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}

```

```

    }
    else
    {
        Log.e(TAG, "Couldn't get json from server.");
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                Toast.makeText(getApplicationContext(),
                    "Συνδεθείτε στα δεδομένα ή στο WiFi",
                    Toast.LENGTH_LONG)
                    .show();
            }
        });
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    StrictMode.ThreadPolicy policy=new
    StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    mainLayout=new LinearLayout(this);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setTheme(R.style.AppTheme);
    setContentView(mainLayout);
    mainLayout.setOrientation(LinearLayout.VERTICAL);
    LinearLayout l=new LinearLayout(this);
    l.setGravity(Gravity.CENTER);
    mainLayout.addView(l);
    LinearLayout bio=new LinearLayout(this);
    bio.setGravity(Gravity.CENTER);
    mainLayout.addView(bio);
    LinearLayout.LayoutParams
    lparams=(LinearLayout.LayoutParams)l.getLayoutParams();
    lparams.width = LinearLayout.LayoutParams.MATCH_PARENT;
    l.setLayoutParams(lparams);
    LinearLayout.LayoutParams
    bparams=(LinearLayout.LayoutParams)bio.getLayoutParams();
    bparams.width = LinearLayout.LayoutParams.MATCH_PARENT;
    bio.setLayoutParams(bparams);
    final ImageView im1=new ImageView(this);
    Bitmap
    bm1=BitmapFactory.decodeResource(getResources(),R.drawable.back1);
    final int width=66;
    final int height=66;
    bm1=bm1.createScaledBitmap(bm1, width,height, true);
    im1.setImageBitmap(bm1);
    l.addView(im1);
    im1.setOnClickListener(new OnClickListener()
    {
        @Override
        public void onClick(View v) {
            onBackPressed();
        }
    });
    im1.setPadding(15,15,15,15);
    im1.setScaleType(ImageView.ScaleType.FIT_START);
    final ImageView im2=new ImageView(this);
    Bitmap
    bm2=BitmapFactory.decodeResource(getResources(),R.drawable.phone);
    final int width1=66;

```

```

        final int height1=66;
        bm2=bm2.createScaledBitmap(bm2, width1,height1, true);
        im2.setImageBitmap(bm2);
        l.addView(im2);
        im2.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder ab = new
AlertDialog.Builder(ProsIntro.this);
                TextView title=new TextView(ProsIntro.this);
                title.setTextColor(Color.rgb(37,45,53));
                title.setPadding(10,10,10,10);
                title.setGravity(Gravity.CENTER);
                title.setTypeface(null, Typeface.ITALIC);
                title.setBackgroundColor(Color.rgb(66,139,202));
                title.setText("Τηλέφωνο Επικοινωνίας Με Βάση Τον
Πίνακα");

                title.setTextSize(20);
                final ArrayList<String> s=new ArrayList<String>();
                Pattern
pattern=Pattern.compile("(69+\\w+) | (26510+\\w+)");
                Matcher matcher=pattern.matcher(introtext);
                while(matcher.find())
                {
                    s.add(matcher.group());
                }
                final ArrayAdapter<String> adapter=new
ArrayAdapter<String>(ProsIntro.this,android.R.layout.simple_list_item_1,s
);
                ab.setNegativeButton("Κλείσιμο", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        dialog.dismiss();
                    }
                });
                ab.setAdapter(adapter, new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        String strName=adapter.getItem(which);
                        Intent i=new Intent(Intent.ACTION_CALL);
                        i.setData(Uri.parse("tel:"+strName));
                        startActivity(i);
                    }
                });
                ab.setCustomTitle(title);
                ab.show();
            }
        });
        im2.setPadding(15,15,15,15);
        im2.setScaleType(ImageView.ScaleType.FIT_END);
        final ImageView im3=new ImageView(this);
        Bitmap
bm3=BitmapFactory.decodeResource(getResources(),R.drawable.email);
        final int width3=66;
        final int height3=66;
        bm3=bm3.createScaledBitmap(bm3, width3,height3, true);

```

```

        im3.setImageBitmap(bm3);
        l.addView(im3);
        im3.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder ab = new
AlertDialog.Builder(ProsIntro.this);
                TextView title=new TextView(ProsIntro.this);
                title.setTextColor(Color.rgb(37,45,53));
                title.setPadding(10,10,10,10);
                title.setGravity(Gravity.CENTER);
                title.setTypeface(null, Typeface.ITALIC);
                title.setBackgroundColor(Color.rgb(66,139,202));
                title.setText("Emails Με Βάση Τον Πίνακα");
                final ArrayList<String> s=new ArrayList<String>();
                Pattern pattern=Pattern.compile("\\b[\\w.%-]+@[-.
.\\w]+\\. [A-Za-z]{2,4}\\b");
                Matcher matcher=pattern.matcher(introtext);
                while(matcher.find())
                {
                    s.add(matcher.group());
                }
                final ArrayAdapter<String> adapter=new
ArrayAdapter<String>(ProsIntro.this,android.R.layout.simple_list_item_1,s
);
                ab.setNegativeButton("Κλείσιμο", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        dialog.dismiss();
                    }
                });
                ab.setAdapter(adapter, new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
which) {
                        final String
email=adapter.getItem(which);
                        AlertDialog.Builder ab = new
AlertDialog.Builder(ProsIntro.this);

                        LinearLayout l=new
LinearLayout(ProsIntro.this);

                        l.setOrientation(LinearLayout.VERTICAL);
                        ab.setView(l);
                        final Dialog d=ab.create();
                        TextView title=new
TextView(ProsIntro.this);

                        title.setTextColor(Color.rgb(37,45,53));
                        title.setPadding(10,10,10,10);
                        title.setGravity(Gravity.CENTER);
                        title.setTypeface(null, Typeface.ITALIC);

                        title.setBackgroundColor(Color.rgb(66,139,202));
                        title.setText("Αποστολή Email σε :
"+email+"");

```



```

        l.addView(title);
        final EditText emailBody=new
EditText(ProsIntro.this);
        emailBody.setHint("Γράψτε το κείμενο του Email
εδώ!!!");

        emailBody.setLines(10);
        l.addView(emailBody);
        final ImageView sendmail=new
ImageView(ProsIntro.this);
        Bitmap
bm=BitmapFactory.decodeResource(getResources(),R.drawable.sendmail);
        final int width=66;
        final int height=66;
        bm=bm.createScaledBitmap(bm, width,height, true);
        sendmail.setImageBitmap(bm);
        l.addView(sendmail);
        sendmail.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent i=new
Intent(Intent.ACTION_SEND);
                i.setType("type/plain");
                i.putExtra(Intent.EXTRA_EMAIL,new
String[]{email});
                i.putExtra(Intent.EXTRA_SUBJECT,
"Αποστολή από την εφαρμογή NursingApp");
                String
myMessage=emailBody.getText().toString();
                i.putExtra(Intent.EXTRA_TEXT,
Html.fromHtml(myMessage));
                try {
startActivity(Intent.createChooser(i,"Αποστολή Email..."));
                }
                catch(android.content.ActivityNotFoundException ex)
                {
                    Toast.makeText(ProsIntro.this,"Δεν υπάρχει εγκατεστημένο πρόγραμμα
διαχείρισης email.",Toast.LENGTH_SHORT).show();
                }
            }
        });
        Button closeButton=new Button(ProsIntro.this);
        l.addView(closeButton);
        closeButton.setText("Κλείσιμο");
        closeButton.setOnClickListener(new
OnClickListener()
        {
            @Override
            public void onClick(View v) {
                d.cancel();
            }
        });
        d.show();
        Window window=d.getWindow();

        window.setLayout(LayoutParams.WRAP_CONTENT,LayoutParams.WRAP_CONTENT);
    }
});

```

```

        ab.setCustomTitle(title);
        ab.show();
    }
});
im3.setPadding(15,15,15,15);
im3.setScaleType(ImageView.ScaleType.FIT_END);
TextView b=new TextView(this);
bio.addView(b);
b.setTextColor(Color.rgb(37,45,53));
b.setText("Λήψη Βιογραφικών");
b.setTextSize(20);
final ImageView im4=new ImageView(this);
Bitmap
bm4=BitmapFactory.decodeResource(getResources(),R.drawable.download);
final int width4=66;
final int height4=66;
bm4=bm4.createScaledBitmap(bm4, width4,height4, true);
im4.setImageBitmap(bm4);
bio.addView(im4);
im4.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v) {
        AlertDialog.Builder ab=new
AlertDialog.Builder(ProsIntro.this);
        TextView title=new TextView(ProsIntro.this);
        title.setTextColor(Color.rgb(37,45,53));
        title.setPadding(10,10,10,10);
        title.setGravity(Gravity.CENTER);
        title.setTypeface(null, Typeface.ITALIC);
        title.setBackgroundColor(Color.rgb(66,139,202));
        title.setText("Βιογραφικά σημειώματα Με Βάση Τον
Πίνακα");

        s1=new ArrayList<String>();
        Pattern pattern=Pattern.compile("(images\\/\\[A-Za-
z]*\\\\/\\[A-Za-z]*_\\[A-Za-z]*_\\[A-Za-z]*.pdf)|(images\\/\\[A-Za-z]*_\\[A-Za-z]*[0-
9]*_gr.pdf)");

        Matcher matcher=pattern.matcher(introtext);
        while(matcher.find())
        {
            s1.add(matcher.group());
        }
        final ArrayAdapter<String> adapter=new
ArrayAdapter<String>(ProsIntro.this,android.R.layout.simple_list_item_1,s
1);

        ab.setNegativeButton("Κλείσιμο", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int
which) {

                dialog.dismiss();
            }
        });
        ab.setAdapter(adapter, new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, final
int which) {

                startLoading("Λήψη...");
                new Thread(new Runnable() {
                    @Override

```

```

        public void run()
        {
            String
url=url2+"/"+adapter.getItem(which);
            DownloadManager.Request request
= new DownloadManager.Request(Uri.parse(url));

            request.setDescription("NursingApp");

            request.setTitle(adapter.getItem(which).toString());

            if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.HONEYCOMB)
            {

                request.allowScanningByMediaScanner();

                request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
            }

            DownloadManager
manager=(DownloadManager) getSystemService(Context.DOWNLOAD_SERVICE);
            manager.enqueue(request);
            runOnUiThread(new Runnable() {
                @Override
                public void run()
                {
                    stopLoading();
                }
            });
        }
    }).start();
    }
});
    ab.setCustomTitle(title);
    ab.show();
}

});
    im4.setPadding(15,0,15,0);
    im4.setScaleType(ImageView.ScaleType.FIT_END);
    prosintro();
    wv=new WebView(this);
    wv.getSettings().setJavaScriptEnabled(true);
    final boolean enableLinks=false;
    wv.setWebViewClient(new WebViewClient() {

        @Override
        public boolean shouldOverrideUrlLoading(WebView view,
String url) {
            return !enableLinks;
        }
    });
    wv.loadData(introtext,"text/html; charset=UTF-8", null);
    mainLayout.addView(wv);
    LinearLayout.LayoutParams
wparams=(LinearLayout.LayoutParams)wv.getLayoutParams();
    Display display = getWindowManager().getDefaultDisplay();
    wparams.width = display.getWidth();
    wparams.height = display.getHeight();
    wv.setLayoutParams(wparams);
}
}

```

Βιβλιογραφία

- [1] https://el.wikipedia.org/wiki/%CE%9A%CE%B9%CE%BD%CE%B7%CF%84%CF%8C_%CF%84%CE%B7%CE%BB%CE%AD%CF%86%CF%89%CE%BD%CE%BF
- [2.1] https://en.wikipedia.org/wiki/History_of_mobile_phones
- [2.2] https://en.wikipedia.org/wiki/List_of_mobile_phone_generations
- [2.3] http://apothetirio.teiep.gr/xmlui/bitstream/handle/123456789/52/tlp_000319a.pdf?sequence=2
- [2.4] <https://www.ired.gr/blog/item/4919-4g-vs-lte-the-differences-explained.html>
- [2.5] https://en.wikipedia.org/wiki/Mobile_operating_system#Current_software_platforms
- [2.6] <https://churchm.ag/timeline-of-mobile-os-evolution-infographic/>
- [2.7] https://en.wikipedia.org/wiki/Mobile_operating_system
- [3] https://el.wikipedia.org/wiki/%CE%99%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1_%CE%B5%CE%BA%CE%B4%CF%8C%CF%83%CE%B5%CF%89%CE%BD_%CF%84%CE%BF%CF%85_Android
- [4.1] <https://gizmodo.com/a-history-of-android-from-cupcake-to-m-1707432419>
- [4.2] <http://www.zougla.gr/technology/phones-camera/article/android-80-oreo-dite-ta-xaraktiristika-tou>
- [5] https://el.wikipedia.org/wiki/IPhone#.CE.9B.CE.B5.CE.B9.CF.84.CE.BF.CF.85.CF.81.CE.B3.CE.B9.CE.BA.CF.8C_.CF.83.CF.8D.CF.83.CF.84.CE.B7.CE.BC.CE.B1_iOS_4.0
- [6.1] https://en.wikipedia.org/wiki/IOS_version_history
- [6.2] https://en.wikipedia.org/wiki/IPhone_OS_1
- [6.3] https://en.wikipedia.org/wiki/IPhone_OS_2
- [6.4] https://en.wikipedia.org/wiki/IPhone_OS_3
- [6.5] https://en.wikipedia.org/wiki/IOS_4
- [6.6] https://en.wikipedia.org/wiki/IOS_5
- [6.7] https://en.wikipedia.org/wiki/IOS_6
- [6.8] https://en.wikipedia.org/wiki/IOS_7
- [6.9] https://en.wikipedia.org/wiki/IOS_8
- [6.10] https://en.wikipedia.org/wiki/IOS_9
- [6.11] https://el.wikipedia.org/wiki/IOS_10
- [6.12] https://el.wikipedia.org/wiki/IOS_11
- [7] https://el.wikipedia.org/wiki/Windows_Phone
- [8] <https://www.moontechnolabs.com/apple-vs-android-comparative-study-2017/>
- [9.1] <http://www.greeceandroid.gr/dev/568-how-to-upload-to-google-play>
- [9.2] <https://clearbridgemobile.com/how-to-submit-an-app-to-the-app-store/>

- [9.3] <https://blogs.msdn.microsoft.com/cdnstudents/2012/11/30/how-to-publish-your-windows-8-app-to-the-store>
- [10.1] <http://www.instructables.com/id/How-To-Setup-Eclipse-for-Android-App-Development/>
- [10.2] <https://www.concretepage.com/ide/eclipse/how-to-install-android-sdk-in-eclipse-kepler>
- [11] <http://hayageek.com/android-http-post-get/>
- [12.1] <http://www.json.org/json-el.html>
- [12.2] <https://el.wikipedia.org/wiki/XML>
- [12.3] <https://fileinfo.com/extension/txt>
- [12.4] <https://el.wikipedia.org/wiki/SQLite>