



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

ΤΕΙ ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή Εργασία

Τζίμας Βασίλειος

Α.Μ: 13978

**“ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΑΠΑΝΤΗΣΗΣ
ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ ΓΙΑ ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ ANDROID”**



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΤΣΟΥΛΟΣ ΙΩΑΝΝΗΣ

**“QUESTIONNAIRE MANAGEMENT AND ANSWERING
APPLICATION FOR ANDROID PHONE”**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, / /2017

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.

2.

3.

© 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 εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή μου κ. Τσούλο Ιωάννη, για την βοήθεια του, την υποστήριξη του και την πολύ καλή συνεργασία που είχαμε καθ' όλο το χρονικό διάστημα εκπόνησης της διπλωματικής αυτής εργασίας. Οι συμβουλές του, οι οδηγίες του αλλά και οι κατευθύνσεις του συνέβαλαν στο μέγιστο για την επιτυχή κατά την άποψη μου ολοκλήρωση της εργασίας αυτής.

Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές του τμήματος, για την πολύτιμη γνώση που μου μεταλαμπάδευσαν και τη βοήθεια που μου πρόσφεραν κάθε φορά που ζητήθηκε.

Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου και τους φίλους μου, που με στήριξαν ηθικά και ψυχολογικά σε ολόκληρη την πορεία μου κατά τη διάρκεια φοίτησης μου στο ΤΕΙ, αλλά και την περίοδο ολοκλήρωσης της πτυχιακής μου εργασίας.

ΠΕΡΙΛΗΨΗ

Στην εποχή μας, η πλειοψηφία του κόσμου χρησιμοποιεί έξυπνα κινητά τηλέφωνα (smartphones). Όλοι έχουν έρθει σε επαφή τουλάχιστον μια φορά με τέτοιου είδους κινητό τηλέφωνο ή έχουν ακούσει για τη διαφορά που έχει, σε σχέση με τα απλά κινητά τηλέφωνα σε λειτουργίες, δυνατότητες και τρόπο χρήσης. Η χρήση του δεν περιορίζεται απλά στην επικοινωνία μέσω κλήσεων και μηνυμάτων με άλλα άτομα, αλλά τείνει να αντικαταστήσει τον ηλεκτρονικό υπολογιστή. Σε όλα τα έξυπνα κινητά τηλέφωνα παρέχονται πληθώρα εφαρμογών, προσφέροντας ευκολία για λειτουργίες που παλιότερα ήταν πιο χρονοβόρες ή δεν μπορούσες να υλοποιήσεις εν κινήσει (πχ πηγαίνοντας στην δουλειά). Οι εφαρμογές αυτές συνεχώς πολλαπλασιάζονται, καλύπτοντας το μεγαλύτερο μέρος των αναγκών των χρηστών.

Στην παρούσα πτυχιακή εργασία, στο πρώτο κεφάλαιο, αναλύονται οι γλώσσες προγραμματισμού, οι δυνατότητες τους, που χρησιμοποιούνται, μερικά παραδείγματα προγραμματισμού της κάθε γλώσσας και ο λόγος που επιλέξαμε την συγκεκριμένη γλώσσα (Java). Στην συνέχεια περιγράφονται τα προβλήματα που αντιμετωπίζει αυτή η εφαρμογή και θα αναφερθούν προγράμματα που λειτουργούν με αντίστοιχο τρόπο. Τέλος, αναφέρεται το λογισμικό που χρησιμοποιήθηκε.

Στο δεύτερο κεφάλαιο, περιγράφονται τα λειτουργικά συστήματα έξυπνων κινητών συσκευών, η ιστορία τους και γίνεται μια αντικειμενική σύγκριση ανάλογα με τα πλεονεκτήματα και μειονεκτήματα του κάθε λογισμικού. Τέλος θα αναλυθούν τα αναγκαία βήματα για την διαδικασία ανεβάσματος της εφαρμογής μέσω της εφαρμογής διανομής εφαρμογών που παρέχει το κάθε λειτουργικό σύστημα.

Στο τρίτο κεφάλαιο, περιγράφονται τα αναγκαία εργαλεία για την ανάπτυξη εφαρμογών android (πολυμέσα, τρόποι αποθήκευσης) και περιέχεται το εγχειρίδιο εγκατάστασης του android plugin στο προγραμματιστικό περιβάλλον eclipse.

Στο τέταρτο κεφάλαιο, γίνεται η παρουσίαση της εφαρμογής που δημιουργήθηκε, τα βήματα που απαιτούνται για την χρήση του και οι δυνατότητες που προσφέρει. Επίσης, αναφέρονται οι τυχόν εξελίξεις που θα μπορούσαν να πραγματοποιηθούν στο μέλλον με σκοπό την βελτίωση της υπάρχουσας εφαρμογής.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ	10
1.1	ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ	10
1.2	ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	11
1.3	ΓΛΩΣΣΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	20
1.4	ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΠΙΛΥΕΙ Η ΕΦΑΡΜΟΓΗ ΜΑΣ	21
1.5	ΠΑΡΟΜΟΙΑ ΠΡΟΓΡΑΜΜΑΤΑ ΛΟΓΙΣΜΙΚΟΥ	21
1.6	ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΟ ΛΟΓΙΣΜΙΚΟ	23
2	ΕΞΥΠΝΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ.....	23
2.1	ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ	23
2.2	ΟΙ ΠΡΩΤΕΣ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ.....	24
2.3	ANDROID	25
2.4	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ANDROID	26
2.5	IOS	27
2.6	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ IOS	28
2.7	WINDOWS PHONE.....	28
2.8	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ WINDOWS PHONE.....	29
2.9	ΣΥΓΚΡΙΣΗ ΜΕΤΑΞΥ ΤΩΝ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΓΙΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ	30
2.10	ΔΙΑΔΙΚΑΣΙΑ ΑΝΕΒΑΣΜΑΤΟΣ ΕΦΑΡΜΟΓΗΣ ΑΝΑ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ .	32
3	ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ANDROID	35
3.1	ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ANDROID PLUGIN ΣΕ ECLIPSE	35
3.2	ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ANDROID ΠΡΟΓΡΑΜΜΑΤΑ	38
4	Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΕΦΑΡΜΟΓΗ	39
4.1	ΔΗΜΙΟΥΡΓΙΑ ΧΡΗΣΤΗ	39
4.2	ΠΑΡΑΔΕΙΓΜΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ	43
4.3	ΚΑΤΕΒΑΣΜΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ	45
4.4	ΕΙΣΟΔΟΣ ΣΤΗΝ ΕΦΑΡΜΟΓΗ	46
4.5	ΔΙΑΔΙΚΑΣΙΑ ΑΠΑΝΤΗΣΗΣ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ.....	47
4.6	ΕΞΕΛΙΞΕΙΣ ΠΟΥ ΘΑ ΜΠΟΡΟΥΣΑΝ ΝΑ ΠΡΑΓΜΑΤΟΠΟΙΗΘΟΥΝ ΣΤΟ ΜΕΛΛΟΝ 50	
5	ΠΑΡΑΡΤΗΜΑ (Ο ΚΩΔΙΚΑΣ).....	51
6	ΒΙΒΛΙΟΓΡΑΦΙΑ	72

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 Γλώσσες Προγραμματισμού	11
Εικόνα 2 Java Programming	20
Εικόνα 3 HTML Programming	20
Εικόνα 4 PHP Programming	20
Εικόνα 5 Facebook logo	22
Εικόνα 6 Twitter logo	22
Εικόνα 7 Viber logo	22
Εικόνα 8 Whatsapp logo	22
Εικόνα 9 Faceboo Gameroom	22
Εικόνα 10 Steam logo.....	22
Εικόνα 11 Viber Desktop logo	22
Εικόνα 12 Η πρώτη εμπορική έκδοση DynaTAC.....	24
Εικόνα 13 Android.....	25
Εικόνα 14 iOS	27
Εικόνα 15 Windows Phone	28
Εικόνα 16 Διαμάχη Λογισμικών Κινητών Τηλεφώνων.....	30
Εικόνα 17 Ποσοστά Προτίμησης Λογισμικού	30
Εικόνα 18 Προτίμηση Λογισμικού ανά Χώρα	31
Εικόνα 19 Play Store.....	32
Εικόνα 20 App Store.....	33
Εικόνα 21 Windows Store	34
Εικόνα 22 Android SDK Manager	35
Εικόνα 23 Open Android Virtua Manager	36
Εικόνα 24 Create Virtual Manager	36
Εικόνα 25 Create Virtual Manager Informations	37
Εικόνα 26 Start Virtual Manager	38
Εικόνα 27 Login and Register to Platform.....	40
Εικόνα 28 Register to Platform	40
Εικόνα 29 Fill out the Form to Register.....	41
Εικόνα 30 Registration Completed.....	41
Εικόνα 31 Data Fail.....	42
Εικόνα 32 Register Fail	42
Εικόνα 33 JOY's Questions and Answers.....	43
Εικόνα 34 Diatrofi's Questions and Answers	43
Εικόνα 35 Taksidia's Questions and Answers	44
Εικόνα 36 Mobile's Questions and Answers	44
Εικόνα 37 Choose Questionnaire	45
Εικόνα 38 Download Questionnaire Manually	45
Εικόνα 39 Login to Application and Download Questionnaire automatically	46
Εικόνα 40 Start Questionnaire	47
Εικόνα 41 First Question	48
Εικόνα 42 Previus Question Available.....	48
Εικόνα 43 Confirm Question Available.....	49
Εικόνα 44 Answers Have Been Sent.....	50

1 ΕΙΣΑΓΩΓΗ

1.1 ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Ο σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η παρουσίαση μιας εφαρμογής για κινητά τηλέφωνα (smartphones) με λειτουργικό σύστημα android και των απαραίτητων διεργασιών που πρέπει να πραγματοποιήσει ο χρήστης για την δημιουργία μιας αντίστοιχης εφαρμογής.

Επιπλέον, παρουσιάζει την δημιουργία ενός εύχρηστου προγράμματος για την δημιουργία και την συμπλήρωση (μέσω κινητού Android) ερωτηματολογίων. Μελετήθηκαν τα περιβάλλοντα (προγραμματιστικά, βάσεων δεδομένων και δημιουργίας ιστοσελίδων) στα οποία θα μπορούσε να υλοποιηθεί.

Για τη δημιουργία της εφαρμογής αυτής που είναι φτιαγμένη για κινητά με λογισμικό android χρειάζεται ένα περιβάλλον που να υποστηρίζει java και πιο συγκεκριμένα java για Android. Για τέτοιου είδους εφαρμογές τα συνηθέστερα περιβάλλοντα είναι: eclipse, Android studio. Κοιτώντας σε βάθος τις επιλογές που υπήρχαν, επέλεξα το περιβάλλον eclipse διότι υπάρχουν περισσότερες πληροφορίες και βοήθειες με αποτέλεσμα να καθιστά την υλοποίηση πιο εύχρηστη και φιλική για τον δημιουργό της εφαρμογής.

Για την δημιουργία της εφαρμογής αυτής, η οποία είναι φτιαγμένη για κινητά με λογισμικό Android, χρειάζεται ένα περιβάλλον που να υποστηρίζει την εκτέλεση κώδικα Java και πιο συγκεκριμένα το πλαίσιο της Java για Android. Για τέτοιου είδους εφαρμογές τα συνηθέστερα περιβάλλοντα είναι το Eclipse και Android studio. Κοιτώντας σε βάθος τις επιλογές που υπήρχαν, επέλεξα το περιβάλλον Eclipse διότι υπάρχουν περισσότερες διαθέσιμες πληροφορίες και βοηθητικό υλικό, με αποτέλεσμα να καθίσταται η υλοποίηση πιο εύχρηστη και φιλική για τον δημιουργό της εφαρμογής.

Για την δημιουργία ιστοσελίδων υπήρχε η επιλογή της χρήσης Wordpress και Joomla, περιβάλλοντα τα οποία προσφέρουν καλύτερο οπτικό περιβάλλον και περισσότερες δυνατότητες, αλλά θεώρησα ότι για το επιθυμητό αποτέλεσμα της συγκεκριμένης εργασίας, δεν χρειαζόταν κάτι τόσο εξεζητημένο. Αυτό είχε σαν αποτέλεσμα να καταλήξουμε στην επιλογή της δημιουργίας ιστοσελίδας χωρίς την χρήση κάποιου τέτοιου εργαλείου. Χρησιμοποίησα κώδικα HTML για το οπτικό αποτέλεσμα, και PHP για την σύνδεση της ιστοσελίδας με την βάση δεδομένων και τους απαραίτητους ελέγχους.

Για την χρήση της ιστοσελίδας χρειαστήκαμε έναν εξυπηρετητή φιλοξενίας (host). Για τη φάση της ανάπτυξης, χρησιμοποιήσαμε το ελεύθερο λογισμικό XAMPP για την δημιουργία ιστοσελίδας σε τοπικό δίκτυο. Έπειτα, θεωρήσαμε ότι επιβάλλεται να χρησιμοποιηθεί κάποιος εξυπηρετητής φιλοξενίας για την εγκατάσταση της εφαρμογής, που μας έδωσε τη δυνατότητα πλοήγησης της ιστοσελίδας χωρίς περιορισμό στο που βρίσκεται ο χρήστης. Χρησιμοποιήθηκε ο εξυπηρετητής 000webhosting.com.

1.2 ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ



Εικόνα 1 Γλώσσες Προγραμματισμού

Με την εφεύρεση του υπολογιστή εμφανίστηκε και η έννοια του προγραμματισμού ηλεκτρονικών υπολογιστών. Ο προγραμματισμός είναι μια γλώσσα επικοινωνίας με τον υπολογιστή και έχει ως σκοπό την υλοποίηση εφαρμογών που εξυπηρετούν τις ανάγκες των χρηστών. Για παράδειγμα, χωρίς την δημιουργία των λειτουργικών συστημάτων με φιλικά περιβάλλοντα εργασίας (Windows, IOS, Linux), η χρήση των υπολογιστών θα ήταν

πολύ δύσκολη έως μη προσιτή για τον μέσο χρήστη.

Αρχικά ο προγραμματισμός στηριζόταν σε ακολουθίες 1 και 0 που αναπαριστούσαν την διέλευση ή μη ρεύματος. Στην συνέχεια όμως, δημιουργήθηκαν οι μεταγλωττιστές (compiler), προγράμματα που διαβάζουν κώδικα γραμμένο σε κάποια γλώσσα προγραμματισμού υψηλού επιπέδου, και τον μεταφράζουν σε ισοδύναμο κώδικα χαμηλότερου επιπέδου, ή αλλιώς σε γλώσσα μηχανής. Αυτό είχε ως αποτέλεσμα να εμφανιστούν περισσότερες πιο εύχρηστες γλώσσες προγραμματισμού, και καθεμία από αυτές να χρησιμοποιείται για συγκεκριμένους σκοπούς ανάλογα με την δυνατότητες της.

ΓΛΩΣΣΕΣ ΧΑΜΗΛΟΥ ΕΠΙΠΕΔΟΥ [1]

- AUTOCODER
- HLA(High Level Assembly)
- FAP(Fortran Assembly Program)
- X#
- MACRO-11
- GNU Assembler

AUTOCODER [2]

Η Autocoder είναι μια γλώσσα που δημιουργήθηκε από την εταιρεία IBM και είναι μια γλώσσα μηχανής. Η ιδιαίτερη δυσκολία στην χρήση αυτής της γλώσσας είναι ότι δεν έδινε δυνατότητα χρήσης βιβλιοθηκών με συναρτήσεις. Αυτό σημαίνει ότι κάθε φορά που χρειαζόταν μια συνάρτηση, έπρεπε να ξανά γραφτεί από τον δημιουργό και δεν υπήρχε η δυνατότητα του ορισμού και κλήσης της οπότε χρειαζόταν.

HLA [3]

Η HLA είναι μια συμβολική γλώσσα που αναπτύχθηκε από τον Randall Hyde και επιτρέπει την χρήση σύνταξης υψηλότερου επιπέδου ώστε να είναι πιο κατανοητή από τους αρχάριους και πιο εύκολη για τους έμπειρους προγραμματιστές. Επιπλέον, υποστηρίζει πιο προηγμένου τύπου δεδομένα. Η σύνταξη της μοιάζει με γλώσσες υψηλού επιπέδου όπως την Pascal και την C. Η γλώσσα αυτή έδωσε την δυνατότητα να ασχοληθεί μεγαλύτερος αριθμός ατόμων με την δημιουργία εφαρμογών.

FAP [4]

Η γλώσσα προγραμματισμού FAP σχεδιάστηκε από τον David Ferguson και είναι μια γλώσσα στην οποία οι εντολές αποτελούνται από μια ακολουθία δυαδικών αριθμών. Αυτό έκανε την χρήση της αρκετά δύσκολη, και είχε ως αποτέλεσμα να είναι υποχρεωτική η ιδιαίτερη εξάσκηση και γνώση σχετικά με τη γλώσσα ώστε να μπορεί κανείς να γράψει κώδικα σε αυτή.

X# [5]

Η X# είναι μια γλώσσα προγραμματισμού χαμηλού επιπέδου, η οποία δημιουργήθηκε από την εταιρεία Cosmos για να χρησιμοποιηθεί στους επεξεργαστές x86. Αρχικά, χρησιμοποιήθηκε από την Cosmos για τον εντοπισμό σφαλμάτων. Η σύνταξη που χρησιμοποιείται σε αυτή τη γλώσσα είναι παρόμοια με αυτή της γλώσσας C.

MACRO-11 [6]

Η Macro-11 είναι μια συμβολική γλώσσα, η οποία δημιουργήθηκε από την Digital Equipment Corporation (DEC). Σχεδιάστηκε για να χρησιμοποιηθεί από την οικογένεια μικροϋπολογιστών PDP-11. Υποστηρίζεται σε όλα τα λειτουργικά συστήματα DEC PDP-11.

ΚΩΔΙΚΑΣ ΣΕ ΓΛΩΣΣΑ MACRO-11

```
.TITLE HELLO WORLD
.MCALL .TTYOUT,.EXIT
HELLO::  MOV #MSG,R1
1$:      MOVB (R1) +, R0
         BEQ  DONE
         .TTYOUT
         BR   1$
DONE:    .EXIT

MSG:     .ASCIZ /Hello, world!
         .END HELLO
```

Το παραπάνω κομμάτι κώδικα εμφανίζει το μήνυμα Hello world στον χρήστη. Όπως φαίνεται, η χρήση της γλώσσας είναι απαιτητική και χρειάζεται ιδιαίτερη δεξιότητα για να γραφτεί κώδικας σε αυτή.

GNU Assembler [7]

Η γλώσσα προγραμματισμού GNU Assembler δημιουργήθηκε από την GNU Project και είναι γνωστή ως GAS ή AS. Χρησιμοποιήθηκε για την δημιουργία του λειτουργικού συστήματος GNU, του Linux Kernel και διαφόρων άλλων λογισμικών. Η κύρια χρησιμότητα της είναι η δημιουργία λειτουργικών συστημάτων. Όπως φαίνεται και στον κώδικα που παρουσιάζεται παρακάτω είναι μια ακόμα γλώσσα με δύσκολη διαχείριση.

ΚΩΔΙΚΑΣ ΣΕ ΓΛΩΣΣΑ GNU Assembler

```
.globl _start

.text
_start:
    movl $len, %edx
    movl $msg, %ecx
    movl $1, %ebx
    movl $4, %eax
    int $0x80

    movl $0, ebx
    movl $1, eax
    int $0x80

.data
Msg
    .ascii "Hello, world! \n"
    len = .-msg
```

Ο παραπάνω κώδικας έχει ως αποτέλεσμα την εμφάνιση του μηνύματος Hello, world στον χρήστη.

ΓΛΩΣΣΕΣ ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ

- FORTRAN
- COBOL
- LISP
- C
- PASCAL

FORTRAN [8]

Η Fortran σχεδιάστηκε στις αρχές της δεκαετίας του 1950 και το όνομα της προέρχεται από τις λέξεις : FORMula TRANslation. Η Fortran σχεδιάστηκε για να βοηθήσει τους επιστήμονες και τους μηχανικούς, και έχει χρησιμοποιηθεί κυρίως για την δημιουργία έργων όπως ο σχεδιασμός γεφυρών και των δομών αεροπλάνου. Επίσης, δίνεται η δυνατότητα στους χρήστες να δημιουργήσουν βιβλιοθήκες με τις λειτουργίες που χρησιμοποιούν πιο συχνά. Αυτό δίνει την δυνατότητα εξοικονόμησης χρόνου στους χρήστες.

COBOL [9]

Η Cobol σχεδιάστηκε από την Grace Hopper στις αρχές του 1960 και είναι η δεύτερη αρχαιότερη γλώσσα προγραμματισμού υψηλού επιπέδου. Είναι ιδιαίτερα δημοφιλής για επιχειρηματικές εφαρμογές που τρέχουν σε μεγάλους υπολογιστές. Τα προγράμματα που γράφονται στην συγκεκριμένη γλώσσα έχουν την ιδιότητα να είναι μεγαλύτερα σε μέγεθος σε σχέση με

προγράμματα που πραγματοποιούν την ίδια λειτουργία γραμμένα σε άλλες γλώσσες. Αυτό την έκανε λιγότερο επιθυμητή στους προγραμματιστές αλλά επιτύγχανε την εύκολη κατανόηση του κώδικα. Παρόλα αυτά, παραμένει η πιο ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού στον κόσμο.

LISP [10]

Η Lisp αποτελείται από μια οικογένεια γλωσσών προγραμματισμού ηλεκτρονικών υπολογιστών που δημιουργήθηκε το 1958. Αρχικά, δημιουργήθηκε για την λύση εφαρμογών που αφορούσαν μαθηματικούς σκοπούς. Στην συνέχεια, χρησιμοποιήθηκε για εφαρμογές που είχαν σχέση με την τεχνητή νοημοσύνη. Σημαντικές ιδιότητες της Lisp είναι το ότι συμπεριλαμβάνει τα δέντρα δομών δεδομένων, και ότι περιέχει αυτόματη διαχείριση αποθήκευσης και δυναμική πληκτρολόγηση.

C [11]

Η γλώσσα προγραμματισμού C είναι μια γλώσσα υψηλού επιπέδου γενικής χρήσης. Αναπτύχθηκε από τον Dennis Ritchie για την ανάπτυξη του λειτουργικού συστήματος UNIX. Όλα τα προγράμματα εφαρμογών της UNIX είναι γραμμένα σε γλώσσα C. Είναι μια γλώσσα αρκετά κατανοητή για τον χρήστη λόγω της χρήσης κατανοητών λέξεων για την κάθε εντολή. Για παράδειγμα, με την εντολή printf, εκτυπώνει ένα μήνυμα στον χρήστη της εφαρμογής. Σημαντικές ιδιότητες της είναι η εύκολη εκμάθηση της, η δομημένη μορφή της, η παραγωγή αποτελεσματικών εφαρμογών, καθώς και το ότι μπορεί να χειριστεί δραστηριότητες χαμηλού επιπέδου.

ΚΩΔΙΚΑΣ ΣΕ ΓΛΩΣΣΑ C

```
main()
{
    printf("Hello World");
}
```

Το παραπάνω κομμάτι κώδικα εμφανίζει το μήνυμα <<Hello World>>

PASCAL [12]

Η Pascal αναπτύχθηκε από τον Niklaus Wirth στο πανεπιστήμιο της Γενεύης, και πήρε το όνομα της προς τιμήν του μαθηματικού και φιλόσοφου Blaise Pascal. Η Pascal είναι μια γλώσσα προγραμματισμού γενικού σκοπού. Παραλλαγές της έχουν χρησιμοποιηθεί για την δημιουργία εφαρμογών κάθε τύπου, όπως παιχνιδιών και ενσωματωμένων συστημάτων. Η Pascal ήταν η κύρια γλώσσα υψηλού επιπέδου που χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος Macintosh. Στις μέρες μας χρησιμοποιείται στα πανεπιστήμια ως εργαλείο εκμάθησης του προγραμματισμού στους μαθητές, επειδή έχει όλα τα στοιχεία του δομημένου προγραμματισμού.

ΚΩΔΙΚΑΣ ΣΕ ΓΛΩΣΣΑ PASCAL

```
Program name;  
Begin  
    Print("Hello World");  
End.
```

Το παραπάνω κομμάτι κώδικα εμφανίζει το μήνυμα <<Hello World>>

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΕΙΣ ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- C++
- JAVA
- DELPHI
- PYTHON

C++ [13]

Η C++ είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού γενικού σκοπού, η οποία αναπτύχθηκε από τον Bjarne Stroustrup το 1983. Η C++ δημιουργήθηκε για να βελτιώσει την ήδη υπάρχουσα γλώσσα προγραμματισμού C. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων και στην συνέχεια ακολούθησαν οι εικονικές συναρτήσεις, η υπερφόρτωση τελεστών και η πολλαπλή κληρονομικότητα. Θεωρείται πολύ δυνατή γλώσσα αν και υστερεί σε βιβλιοθήκες σε σχέση με άλλες αντικειμενοστραφείς γλώσσες. Ωστόσο, είναι πολύ γρήγορη και έχει πολλές δυνατότητες. Κύριο χαρακτηριστικό της είναι η εύκολη κατανόηση των εντολών που χρησιμοποιούνται. Αυτό επιτυγχάνεται με την χρήση λέξεων που χαρακτηρίζουν την κάθε εντολή και είναι εύκολο να κατανοηθεί με την εντολές που περιγράφονται παρακάτω.

ΚΩΜΜΑΤΙΑ ΚΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ C++

```
Int main()  
{  
    Cout<< "Hello world!!!";  
    Return 0;  
}
```

Το παραπάνω κομμάτι κώδικα εμφανίζει το μήνυμα <<Hello world!!!>>

```
Int main()  
{  
    Int i, x, p;  
    Cout<<"dwse arithmo gia ton οποιο thes na upologistei to paragontiko";  
    Cin>>x;  
    p=1;  
    For(i=0; i<=x; i++)
```

```
{
    p=p*i;
}
Cout<<"το παραγοντικο του" <<x <<"einai :'" <<p;
Return 0;
}
```

Ο παραπάνω κώδικας υπολογίζει και εμφανίζει το παραγοντικό του αριθμού που δίνει ο χρήστης.

JAVA [14]

Η Java σχεδιάστηκε από τον James Gosling και τον Sun Microsystems το 1995 και πήρε το όνομα της από το νησί Ιάβα. Είναι η γλώσσα που χρησιμοποιείται κυρίως σε εφαρμογές διαδικτύου. Είναι η πρώτη γλώσσα που μπορούσε να συμπεριλάβει ήχο και κίνηση σε μια ιστοσελίδα. Επιπλέον, δίνει την δυνατότητα στους χρήστες να αλληλεπιδρούν με τις ιστοσελίδες. Επίσης, χρησιμοποιείται στην δημιουργία παιχνιδιών και στην ανάπτυξη εφαρμογών για συνομιλία μέσω internet (skype). Η γλώσσα αυτή χρησιμοποιείται επίσης για την κατασκευή εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα android και IOS.

ΚΩΜΑΤΙΑ ΚΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ JAVA

```
class HelloWorld {
    public static void main (String args[]) {
        System.out.println("Hello World!");
    }
}
```

Το παραπάνω κομμάτι κώδικα εμφανίζει στην οθόνη το μήνυμα <Hello World!>

```
class FahrToCelsius {
    public static void main (String args[]) {
        int fahr, celsius;
        int lower, upper, step;
        lower = 0;
        upper = 300;
        step = 20;
        fahr = lower;
        while (fahr <= upper) {
```



```

celsius = 5 * (fahr-32) / 9;
System.out.print(fahr);
System.out.print(" ");
System.out.println(celsius);
fahr = fahr + step;
} }

}

```

Το παραπάνω κομμάτι κώδικα υπολογίζει έναν πίνακα θερμοκρασιών Fahrenheit αν του δώσουμε θερμοκρασίες Celsius.

Για περισσότερα παραδείγματα στην πηγή [15].

DELPHI

Η Delphi είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία Embarcadero και αναπτύχθηκε από μια ομάδα με επικεφαλής τον Larry Tesler, σε συνεννόηση με τον εφευρέτη της Pascal, Niklaus Wirth. Η Delphi έχει ως βάση την Pascal και προσφέρει επιπλέον την δυνατότητα δημιουργίας κλάσεων και αντικειμένων. Χρησιμοποιήθηκε για να υποστηρίξει την MacApp (επεκτάσιμη εφαρμογή του Macintosh). [16]

KOMMATI KΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ DELPHI

```

Program TestLeft;
Var
    x: integer;
    xSB: StringBuilder;
begin
    writeln ( ' Start' );
    xSB:=StringBuilder.Create ( 'Hello CrossTalk!!!' );
    writeln ( ' Length: ' + IntToStr(xSB.Length));
    xSB.append ( ' More' );
    writeln( ' Length: ' + IntToStr(xSB.Length));
    writeln(xSB.ToString);
    writeln( ' end' );
    readln;

```

```
End.
```

Το παραπάνω πρόγραμμα έχει ως λειτουργία τον υπολογισμό των γραμμάτων μιας πρότασης και τον καινούργιο υπολογισμό στην περίπτωση προσθήκης και άλλης λέξης.[17]

PYTHON [18]

Η Python είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού γενικού σκοπού, η οποία δημιουργήθηκε από τον Guido van Rossum. Λόγω της εύκολης σύνταξης, την κάνει να είναι ιδανική για κάποιον που προσπαθεί να μάθει προγραμματισμό για πρώτη φορά.

KOMMATI KΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ PYTHON

Για την εμφάνιση μηνύματος Hello World ο κώδικας που χρησιμοποιείται είναι:

```
Print("Hello World")
```

Ενώ για την εμφάνιση των αριθμών από το 1 μέχρι το 10:

```
For a in range(1,10):  
    Print(a)
```

ΓΛΩΣΣΕΣ ΟΠΤΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Visual Basic
- C#

Visual Basic [19]

Η Visual Basic είναι μια γλώσσα προγραμματισμού τρίτης γενιάς που δημιουργήθηκε από την Microsoft, ακολουθούμενη από συμβάντα. Προέρχεται από την Basic και επιτρέπει την ταχεία ανάπτυξη εφαρμογών με γραφικό περιβάλλον. Έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και τον χειρισμό. Για την δημιουργία του οπτικού αποτελέσματος χρησιμοποιείται η τεχνική «σύρε και άσε» (drag-and-drop), δηλαδή χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (κουμπιά, πλαίσια κειμένου) στην φόρμα.

C# [20]

Η C# είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε από την Microsoft με σκοπό να συνδυάσει την υπολογιστική ισχύ της C++ με την ευκολία προγραμματισμού της Visual Basic. Ο κώδικας της συγκεκριμένης γλώσσας βασίζεται στην C++ αλλά έχει κοινά χαρακτηριστικά με την Java. Η γλώσσα αυτή χρησιμοποιείται επίσης για την κατασκευή εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα Windows Phone.

Περισσότερες γλώσσες προγραμματισμού και πληροφορίες σχετικά με αυτές βρίσκονται στην πηγή [21]

ΓΛΩΣΣΕΣ ΔΗΜΙΟΥΡΓΙΑΣ ΙΣΤΟΣΕΛΙΔΩΝ

- **HTML**
- **PHP**
- JavaScript
- CSS

HTML

Η γλώσσα προγραμματισμού HTML επινοήθηκε το 1989 από τον Tim Berners Li, οποίος εργαζόταν στο Cern και πήρε το όνομα της από τα αρχικά HyperTextMarkup Language και χρησιμοποιείται για την δημιουργία ιστοσελίδων. Η γλώσσα προγραμματισμού HTML δίνει την δυνατότητα στον χρήστη να συνδυάσει την ιστοσελίδα και με κώδικα php, JavaScript, CSS. [51]

ΚΟΜΜΑΤΙ ΚΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ HTML

```
<HTML>
  <HEAD>
    <TITLE>Hello World in HTML</TITLE>
  </HEAD>
  <BODY>
    <CENTER><H1>Hello World!</H1></CENTER>
  </BODY>
</HTML>
```

Ο παραπάνω κώδικας εμφανίζει το μήνυμα “Hello World” σε μια ιστοσελίδα με όνομα Hello World in HTML[52]

PHP

Η γλώσσα προγραμματισμού php είναι μια γλώσσα που χρησιμοποιείται για την δημιουργία ιστοσελίδων web με δυναμικό περιεχόμενο (δηλαδή το αποτέλεσμα που εμφανίζεται δεν είναι πάντα το ίδιο αποτέλεσμα). Η php δημιουργήθηκε από τον φοιτητή Rasmus Lerdorf το 1994 και για την δημιουργία της στηρίχθηκε στην γλώσσα προγραμματισμού C. [53]

ΚΟΜΜΑΤΙ ΚΩΔΙΚΑ ΣΕ ΓΛΩΣΣΑ PHP

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Ο παραπάνω κώδικας εμφανίζει το μήνυμα “hello world” σε μια ιστοσελίδα με όνομα PHP Test [54]. Όποιος διαπιστώνουμε από τον παραπάνω κώδικα, η γλώσσα php είναι συνδεδεμένη με την γλώσσα προγραμματισμού HTML.

JAVASCRIPT

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, μετονομάστηκε σε LiveScript και τελικά σε JavaScript επειδή η δημιουργία της στηρίχθηκε στην γλώσσα προγραμματισμού Java. [55]

CSS

Η γλώσσα προγραμματισμού CSS (Cascading Style Sheets), είναι μια γλώσσα που κύριος σκοπός της είναι ο έλεγχος της εμφάνισης μια ιστοσελίδας. Η γλώσσα αυτή παρουσιάστηκε τον Μάρτιο του 1998. [56]

1.3 ΓΛΩΣΣΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

JAVA



Εικόνα 2 Java Programming

Η γλώσσα που χρησιμοποιήθηκε για την συγγραφή του κώδικα της εφαρμογής για είναι η Java. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, η γλώσσα αυτή χρησιμοποιείται συνήθως για την δημιουργία εφαρμογών για κινητά τηλέφωνα με λειτουργικό σύστημα android.

HTML



Εικόνα 3 HTML Programming

Για την δημιουργία του οπτικού αποτελέσματος της ιστοσελίδας χρησιμοποιήθηκε η γλώσσα προγραμματισμού HTML. Η γλώσσα προγραμματισμού php μας δίνει την δυνατότητα να αλλάξουμε χρώμα στο φόντο και στην γραμματοσειρά της ιστοσελίδας μας, όπως επίσης και να δημιουργήσουμε φόρμες και κουμπιά τα οποία είναι χρήσιμα στην σύνδεση και εγγραφή του χρήστη.

PHP



Εικόνα 4 PHP Programming

Για την σύνδεση και εμφάνιση δεδομένων από την βάση δεδομένων στην ιστοσελίδα χρησιμοποιήθηκε η γλώσσα προγραμματισμού php. Επίσης χρησιμοποιήθηκε για την δημιουργία ελέγχων και επαναλήψεων εντός της ιστοσελίδας.

1.4 ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΕΠΙΛΥΕΙ Η ΕΦΑΡΜΟΓΗ ΜΑΣ

Κατά την δημιουργία και την απάντηση ενός ερωτηματολογίου μπορεί να παρουσιαστούν προβλήματα, κάποια εκ των οποίων, λύνονται μέσω του συγκεκριμένου τρόπου ανάπτυξης και απάντησης ερωτηματολογίου. Τα προβλήματα που επιλύονται από την ανάπτυξη της εφαρμογής μας είναι

1) Ενώ το ερωτηματολόγιο έχει τεθεί προς απάντηση, υπάρχει η πιθανότητα ο δημιουργός του να θελήσει να προσθέσει, να αφαιρέσει ή να τροποποιήσει κάποιες ερωτήσεις ή απαντήσεις. Μέσω της ιστοσελίδας διαχείρισης των ερωτηματολογίων, ο χρήστης διαχειριστής του ερωτηματολογίου μπορεί να το επεξεργαστεί και να το μορφοποιήσει.

2) Δίνεται η δυνατότητα στον χρήστη να απαντήσει συγχρόνως περισσότερα από ένα ερωτηματολόγια γρήγορα και εύκολα. Κάθε φορά που ο χρήστης ενεργοποιεί την εφαρμογή και συμπληρώνει το όνομα του ερωτηματολογίου που επιθυμεί να απαντήσει, αυτόματα η εφαρμογή κατεβάζει το αρχείο που χρειάζεται. Επομένως, αν ο χρήστης κλείσει την εφαρμογή και την ενεργοποιήσει συμπληρώνοντας άλλο όνομα ερωτηματολογίου, θα έχει την δυνατότητα να απαντήσει το επόμενο ερωτηματολόγιο.

3) Ελαχιστοποίηση πιθανότητας απώλειας των δεδομένων. Τα δεδομένα αποθηκεύονται αυτόματα στην βάση δεδομένων της ιστοσελίδας παρέχοντας ασφάλεια δεδομένων. Τα δεδομένα είναι προσβάσιμα μόνο από τον διαχειριστή της εφαρμογής.

4) Ευκολότερη διαχείριση των δεδομένων (απαντήσεων) που έχουν δώσει οι χρήστες από τον δημιουργό του ερωτηματολογίου. Τα δεδομένα έχουν αποθηκευτεί σε βάση δεδομένων SQL, δίνοντας την δυνατότητα στον δημιουργό εύκολα και γρήγορα με την χρήση ερωτημάτων (queries), να βγάλει συμπεράσματα σχετικά με το μέσο όρο, το αθροιστικό αποτέλεσμα μιας συγκεκριμένης απάντησης και εύρεση της απάντησης που έδωσε ένας συγκεκριμένος χρήστης.

5) Αποφυγή πολλαπλών απαντήσεων από τον ίδιο χρήστη για το ίδιο ερωτηματολόγιο. Όσες φορές και να χρησιμοποιηθεί η εφαρμογή με το ίδιο όνομα ερωτηματολογίου, η βάση δεδομένων θα συγκρατήσει τις απαντήσεις που έδωσε ο συγκεκριμένος χρήστης την πρώτη φορά συμπλήρωσης του ερωτηματολογίου.

1.5 ΠΑΡΟΜΟΙΑ ΠΡΟΓΡΑΜΜΑΤΑ ΛΟΓΙΣΜΙΚΟΥ

Αναζητώντας στο διαδίκτυο ή σε προγράμματα ή σε εφαρμογές διανομής εφαρμογών (Play Store, App Store, Windows Store) μπορεί κανείς να διαπιστώσει ότι υπάρχουν πολλές εφαρμογές που λειτουργούν με παρόμοιο τρόπο με την εφαρμογή που παρουσιάζεται στην συγκεκριμένη πτυχιακή εργασία. Ο τρόπος λειτουργίας της είναι:

- Κατέβασμα πληροφοριών από την βάση δεδομένων μιας ιστοσελίδας.
- Επεξεργασία και προβολή των πληροφοριών
- Ενημέρωση της βάσης δεδομένων της ιστοσελίδας από την οποία αποκτήθηκαν τα δεδομένα.

Υπάρχουν πολλές εφαρμογές που λειτουργούν με αντίστοιχο τρόπο και υπάρχουν στα περισσότερα έξυπνα τηλέφωνα ή ηλεκτρονικούς υπολογιστές

Αντίστοιχες εφαρμογές σε κινητά τηλέφωνα



Facebook: η εφαρμογή διαχείρισης και προβολής πληροφορίας της ιστοσελίδας www.facebook.com.

Εικόνα 5 Facebook logo

Πραγματοποιώντας σύνδεση με τα ίδια στοιχεία που της ιστοσελίδας, δίνεται η δυνατότητα προβολής και επεξεργασίας των στοιχείων που έχουν ήδη δοθεί (μηνύματα, εικόνες, likes). Προκειμένου οι πληροφορίες να είναι ίδιες και στην ιστοσελίδα και την εφαρμογή, υπάρχει μια κοινή βάση δεδομένων για την αποθήκευση των πληροφοριών.

Twitter: η εφαρμογή διαχείρισης και προβολής πληροφορίας της ιστοσελίδας www.twitter.com. Όπως και στην εφαρμογή του facebook, πραγματοποιώντας σύνδεση έχεις πρόσβαση στα στοιχεία που έχουν δοθεί και στην ιστοσελίδα.



Εικόνα 6 Twitter logo

Viber: μια εφαρμογή επικοινωνίας μέσω διαδικτύου, η οποία δίνει την δυνατότητα στους χρήστες της να στέλνουν μηνύματα και να



Εικόνα 7 Viber logo

πραγματοποιούν κλήσεις δωρεάν. Η εφαρμογή αυτή είναι διαθέσιμη και για ηλεκτρονικούς υπολογιστές. Προφανώς, υπάρχει μια κοινή βάση δεδομένων, η οποία ενημερώνεται ταυτόχρονα και από τις δυο εφαρμογές.



Εικόνα 8 Whatsapp logo

Whatsapp: παρόμοια εφαρμογή με το viber. Δίνει την δυνατότητα δωρεάν επικοινωνίας μέσω μηνυμάτων και κλήσεων. Η διαφορά με την εφαρμογή του viber είναι ότι δεν υπάρχει εφαρμογή μέσω εφαρμογής για υπολογιστές, αλλά υπάρχει η ιστοσελίδα <https://web.whatsapp.com> μέσω της οποίας μπορείς να κάνεις τις ίδιες λειτουργίες που σου προσφέρει και η εφαρμογή του κινητού τηλεφώνου.

Αντίστοιχες Εφαρμογές για Ηλεκτρονικούς Υπολογιστές.



Facebook Gameroom: μέσω τις συγκεκριμένης εφαρμογής μπορείς να έχεις πρόσβαση στα παιχνίδια που χρησιμοποιείς στην ιστοσελίδα του Facebook.

Εικόνα 9 Faceboo Gameroom

Steam: πλατφόρμα ψηφιακής διανομής, διαχείρισης ψηφιακών δικαιωμάτων και κοινωνικής δικτύωσης. Η Εφαρμογή αυτή, παρέχει στον χρήστη εγκαταστάσεις και αυτόματες ενημερώσεις παιχνιδιών. Η εφαρμογή χρησιμοποιεί τα στοιχεία την ιστοσελίδας <http://store.steampowered.com/>, δίνοντας παρόμοιες δυνατότητες αλλά πιο εύκολη χρήση (ειδικά για τις ενημερώσεις των παιχνιδιών που χρησιμοποιείς).



Εικόνα 10 Steam logo

Viber Desktop: Όπως και η εφαρμογή για κινητά τηλέφωνα δίνει την δυνατότητα δωρεάν επικοινωνίας με την χρήση όμως Ηλεκτρονικού Υπολογιστή. Απαραίτητη προϋπόθεση η ύπαρξη της εφαρμογής και σε κινητό τηλέφωνο για την επιβεβαίωση του αριθμού.



Εικόνα 11 Viber Desktop logo

1.6 ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΟ ΛΟΓΙΣΜΙΚΟ

Για την ανάπτυξη της εφαρμογής για την δημιουργία και απάντηση ερωτηματολογίων χρησιμοποιήθηκαν:

1. Eclipse με το Android SDK για την ανάπτυξη της εφαρμογής για κινητά με λογισμικό Android.
2. Κώδικας HTML, PHP και SQL για την ανάπτυξη της ιστοσελίδας για την δημιουργία χρηστών, ερωτηματολογίων και την διαχείριση τους.
3. Βάση δεδομένων που παρέχει ο εξυπηρετητής φιλοξενίας (host, 000webhosting.com) για την δημιουργία πινάκων για τα δεδομένα που χρειαζόμαστε (χρήστες, ονόματα ερωτηματολογίων, ερωτήσεις και απαντήσεις)

2 ΕΞΥΠΝΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ

Στο κεφάλαιο αυτό θα γίνει μια ιστορική αναδρομή στα έξυπνα κινητά τηλέφωνα (smartphones). Θα αναλυθούν τα λογισμικά τα οποία χρησιμοποιούνται σήμερα και πώς εξελίχθηκαν με το πέρασμα του χρόνου. Θα γίνει μια σύγκριση μεταξύ τους, αναφέροντας τους κύριους λόγους προτίμησης και αποφυγής κάθε λογισμικού. Τέλος, θα περιγραφεί ο τρόπος με τον οποίον μπορούν να εγκατασταθούν οι εφαρμογές σε κάθε λογισμικό.

2.1 ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

Κινητό τηλέφωνο ονομάζεται το τηλέφωνο, που δεν εξαρτάται από φυσική καλωδιακή σύνδεση με δίκτυο τηλεφωνίας, ούτε από κάποια τοπική ασύρματη συσκευή εκπομπής ραδιοφωνικού σήματος χαμηλής συχνότητας. Τα κινητά χρησιμοποιούν τεχνολογία κυψελών και εκπέμπουν σε υψηλές συχνότητες. Για αυτή την εκπομπή και λήψη των σημάτων, χρησιμοποιείται αποκλειστικά ψηφιακή τεχνολογία με κωδικοποίηση. Την σημερινή εποχή όλο και περισσότεροι άνθρωποι χρησιμοποιούν κινητά τηλέφωνα. Αυτές τις συσκευές μπορούμε να τις βρούμε σε διάφορα μεγέθη, χρώματα και δυνατότητες ανάλογα με της προτιμήσεις μας. Τα κινητά τηλέφωνα είναι ένα από τα πολλά δημιουργήματα των τηλεπικοινωνιών και δίνουν την δυνατότητα στο χρήστη να επικοινωνήσει με άλλα άτομα χωρίς να υπάρχει περιορισμός στην τοποθεσία που βρίσκεται, καθώς πλέον στα περισσότερα σημεία παρέχεται σήμα κινητής τηλεφωνίας. Αυτή η επικοινωνία γίνεται με μια απέραντη σειρά δικτύων που συνδέουν τις συσκευές, όπως τα δημόσια τηλεφωνικά δίκτυα, τα δίκτυα υπολογιστών, τα δίκτυα κινητών τηλεφώνων ή ακόμα το δίκτυο που έχει η τηλεόραση και το ραδιόφωνο. Τα κινητά τηλέφωνα παρέχουν κι άλλες δυνατότητες, όπως την αποθήκευση πληροφοριών είτε αυτές οι πληροφορίες είναι εικόνες τραγούδια, βίντεο, μουσική είτε αλληλογραφία. Ο σύγχρονος τρόπος ζωής επιβάλλει την χρήση τέτοιων συσκευών. [22]

2.2 ΟΙ ΠΡΩΤΕΣ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ



Εικόνα 12 Η πρώτη εμπορική έκδοση DynaTAC

Η ιστορία των κινητών τηλεφώνων ξεκίνησε αμέσως μετά τον β' παγκόσμιο πόλεμο, όπου η Σουηδία, η Φινλανδία και η Αμερική προσπαθούσαν να δημιουργήσουν το πρώτο κινητό τηλέφωνο. Στις 3 Απριλίου του 1973 στην Νέα Υόρκη ο δόκτωρ Μάρτιν Κούπερ της Motorola είχε στα χέρια του το πρώτο κινητό τηλέφωνο το οποίο ήταν λίγο μεγάλο, και έμοιαζε με φορητό ασύρματο. Είχε ύψος 25 εκατοστά και βάρος γύρω στο ένα κιλό, και είχε το κωδικό όνομα Motorola DynaTAC. Η Bell το 1978 κατασκεύασε το πρώτο δοκιμαστικό δίκτυο τηλεφωνίας το οποίο ήταν αναγκαίο για την εξέλιξη και την εμπορική εκμετάλλευση του κινητού. Το πρώτο αυτοματοποιημένο δίκτυο κινητής τηλεφωνίας λειτούργησε περίπου στις αρχές του 1980 στην Σκανδιναβία.

Τα κινητά τηλέφωνα μέχρι και το 1990 ήταν αρκετά μεγάλα με αποτέλεσμα να καθιστούν δύσκολη μεταφορά τους. Τα πρώτα κινητά τηλέφωνα που κυκλοφόρησαν τότε ήταν τα λεγόμενα πρώτης γενιάς (1G). Μετά το 1990 άρχισε η απογείωση της παραγωγής των κινητών τηλεφώνων με την ψηφιοποίηση των δικτύων (GSM) και των συσκευών. Οι συσκευές μίκρυναν αρκετά, πλησιάζοντας το μέγεθος της παλάμης του χεριού, αλλά η μεταφορά τους παρέμενε δύσκολη. Στην συνέχεια περάσαμε στα κινητά της δεύτερης γενιάς (2G) με τα όποια υπήρχε η δυνατότητα αποστολής μηνυμάτων και φωτογραφιών. Στις αρχές του 21ου αιώνα ήρθαν τα κινητά τρίτης γενιάς (3G) με απεριόριστες δυνατότητες που εξελίσσονται συνεχώς. Εκτός από την εξωτερική τους εμφάνιση τα κινητά άλλαξαν και ως προς το λειτουργικό τους σύστημα. Τα πρώτα κινητά είχαν ασπρόμαυρα γραφικά. Στην συνέχεια, η εταιρίες άρχισαν να παρουσιάζουν διαφορετικά λογισμικά έχοντας ως αποτέλεσμα την καλύτερη οπτική εμφάνιση της πληροφορίας. [23]

Η εξέλιξη των λειτουργικών συστημάτων για κινητά τηλέφωνα ακολούθησε την εξής πορεία [24]:

- Το 1992 εισήχθη το πρώτο λειτουργικό σύστημα με εφαρμογές για διευθύνσεις, ημερομηνίες και υπενθύμιση πληροφοριών από την εταιρία PALM της RIM.
- Η ίδια εταιρία τον Οκτώβριο του 1997 πρόσθεσε την εφαρμογή του e-mail.
- Το Σεπτέμβριο του 1998, ξανά η PALM, αναβαθμίζει το λειτουργικό της σύστημα προσθέτοντας την υποστήριξη περισσότερων χρωμάτων (8 bit)
- Ακολουθεί η Microsoft εισάγοντας το 2000 το Pocket PC 2000.
- Το 2001 η Microsoft προσθέτει την εφαρμογή MSN messenger και το Media player στο Pocket PC 2002.
- Η PALM την ίδια χρονιά βελτιώνει την υποστήριξη χρωμάτων της οθόνης (16 bit).
- Την επόμενη χρονιά, η PALM εισάγει το Bluetooth.
- Το 2003, Η Microsoft εισάγει και αυτή το Bluetooth και την εφαρμογή του Internet Explorer.
- Το 2004, η PALM, επιτυγχάνει την συνδεσιμότητα μέσω Bluetooth και Wi-Fi στο λειτουργικό σύστημα Palm Cobalt.
- Τον Μάιο της επόμενης χρονιάς, η Microsoft εισάγει την εφαρμογή GPS (Global Positioning Systems).

- Τον Οκτώβριο του ίδιου έτους εισάγεται στην αγορά από την εταιρία BlackBerry το BlackBerry OS 4.1.
- Το 2007 εμφανίζεται το πρώτο κινητό τηλέφωνο της Apple, με την ονομασία iPhone.
- Το 2008, κυκλοφορεί το λειτουργικό σύστημα Android από την Google και το λειτουργικό σύστημα Symbian από την Nokia.
- Το 2011 η Nokia επιτυγχάνει συμφωνία με την Microsoft για την χρησιμοποίηση του λειτουργικού συστήματος Windows Phone και το 2014 τερματίζει την υποστήριξή της στο λειτουργικό σύστημα Symbian.

Στην σημερινή εποχή, η κατάσταση είναι διαφορετική καθώς κάποιες εταιρίες (Google, Apple, Windows, κ.α.) έχουν δημιουργήσει κάποια λογισμικά, τα οποία εξελίσσονται, και οι εταιρίες παραγωγής κινητών τα εγκαθιστούν στα κινητά τηλέφωνα (smartphones). Παλαιότερα κάθε εταιρία παραγωγής κινητών τηλεφώνων δημιουργούσε το δικό της λειτουργικό το οποίο χρησιμοποιούσε στα κινητά τηλέφωνα. Ωστόσο σήμερα βλέπουμε το ίδιο λειτουργικό σύστημα σε διαφορετικές εταιρίες κινητών. Ο λόγος για τον οποίο επικράτησε αυτή η τάση, είναι για την ευκολία των χρηστών. Ο χρήστης μπορεί να χειρίζεται διαφορετικά κινητά τηλέφωνα χωρίς να κάνει προσπάθεια να μάθει το νέο λειτουργικό. Επίσης, το λειτουργικό σύστημα δεν επηρεάζεται από την οθόνη ή την εξωτερική εμφάνιση της συσκευής. Αντιθέτως, ο επεξεργαστής και η μνήμη τυχαίας προσπέλασης (RAM) επηρεάζονται από το λειτουργικό. Κάποια λειτουργικά συστήματα είναι πιο αποδοτικά σε συγκεκριμένες λειτουργίες και έχουν την δυνατότητα να χρησιμοποιούν λιγότερους πόρους με αποτέλεσμα να υπάρχει περισσότερη ελεύθερη ισχύ για τις υπόλοιπες λειτουργίες. Στην Ελλάδα τα κινητά τηλέφωνα έκαναν αισθητή την παρουσία τους το 1992. Οι πρώτες εταιρίες κινητής τηλεφωνίας ήταν η Vodafone και η Wind (με τις σημερινές τους ονομασίες). Το 2006 η Ελλάδα έγινε μια από τις πρώτες χώρες σε χρήση κινητών τηλεφώνων. Παρακάτω θα αναφερθούν περισσότερες πληροφορίες για τα λογισμικά που των εταιριών που κατέχουν το μεγαλύτερο μέρος της αγοράς σήμερα. [25]

2.3 ANDROID



Εικόνα 13 Android

Το Android είναι ένα λειτουργικό σύστημα για κινητά τηλέφωνα ή αλλιώς smartphones, τα οποία είναι κατασκευασμένα με βάση το λειτουργικό σύστημα Linux. Αυτός είναι και ο κύριος λόγος που η πλατφόρμα Android είναι ανοιχτού κώδικα. Το λογισμικό Android κατασκευάστηκε από την εταιρία Google, και είναι μια από τις πιο γρήγορα αναπτυσσόμενες πλατφόρμες αποτελώντας έτσι το μεγαλύτερο μέρος της αγοράς κινητών συσκευών με αυτό το λογισμικό. Το λειτουργικό και τα προγράμματα Android μπορούν να εισαχθούν σε πολλούς διαφορετικούς κατασκευαστές κινητών συσκευών. Σκοπός της σχεδίασης του λογισμικού αυτού είναι η παραμετροποίηση από τον χρήστη, ανάλογα με τις ανάγκες του.

Το λειτουργικό Android, όπως προαναφέρθηκε, βασίζεται στον πυρήνα του λειτουργικού συστήματος Linux και πλέον αναπτύσσεται από την Open Handset Alliance (OHA). Δημιουργήθηκε στις 5 Νοεμβρίου του 2007 κατόπιν συνεργασίας κορυφαίων εταιρειών στο χώρο των τηλεπικοινωνιών όπως η Google, η Motorola, η HTC, η Samsung, η LG και άλλες. Το μεγαλύτερο μέρος του κώδικα του Android δημοσιεύθηκε από την Google υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού, γεγονός που δίνει το σημαντικό πλεονέκτημα αυτού του λειτουργικού. Με βάση αυτή την άδεια κάθε προγραμματιστής έχει τη δυνατότητα να σχεδιάσει εφαρμογές για το συγκεκριμένο λειτουργικό σύστημα με τη χρήση της γλώσσας προγραμματισμού Java. Στο επόμενο κεφάλαιο θα δούμε την ιστορική αναδρομή των εκδόσεων Android και τις διαφορές μεταξύ τους.

2.4 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ ANDROID

- Η πρώτη έκδοση (1.0) κυκλοφόρησε στις 23 Σεπτεμβρίου το 2008 με επίπεδο API 1.
- Η έκδοση 1.1 κυκλοφόρησε στις 9 Φεβρουαρίου του 2009 με επίπεδο API 2. [26]

Από τον Απρίλιο του 2009, οι εκδόσεις android έχουν θέμα την ζαχαροπλαστική στην ονομασία τους

- Η έκδοση 1.5 με κωδικό όνομα Cupcake κυκλοφόρησε στις 27 Απριλίου του 2009 με επίπεδο API 3.
 - Η έκδοση 1.6 με κωδικό όνομα Donut κυκλοφόρησε στις 15 Σεπτεμβρίου του 2009 με επίπεδο API 4 με επιπλέον χαρακτηριστικά, την υποστήριξη CDMA smartphones, υποστήριξη μεγαλύτερης οθόνης, προσθήκη δείκτη χρήσης της μπαταρίας και μια μηχανή μετατροπής κειμένου σε ομιλία. [27]
 - Οι εκδόσεις 2.0- 2.1 με κωδικό όνομα Eclair κυκλοφόρησαν στις 26 Οκτωβρίου του 2009 με επίπεδα API από 5 έως 7 προσθέτοντας υποστήριξη για επικοινωνία κοντινού πεδίου (NFC), ικανότητα αναζήτησης αποθηκευμένων μηνυμάτων SMS και MMS, υποστήριξη Exchange για την εφαρμογή e-mail και βελτίωσε το Google Maps 3.1.2. [28]
 - Οι εκδόσεις 2.2- 2.2.3 με κωδικό όνομα Froyo κυκλοφόρησαν στις 20 Μαΐου 2010 με επίπεδο API 8 με κυριότερες αλλαγές την πρόσδεση USB και την λειτουργία Wi-Fi hotspot. [29]
 - Οι εκδόσεις 2.3- 2.3.7 με κωδικό όνομα Gingerbread κυκλοφόρησαν στις 6 Δεκεμβρίου 2010 με επίπεδα API 9- 10. Οι εκδόσεις αυτές εισήγαγαν το Session Initiation Protocol (SIP) που χρησιμοποιείται στην τηλεφωνία VoIP τηλεφωνία μέσω internet.[30]
 - Οι εκδόσεις 3.0- 3.2.6 με κωδικό όνομα Honeycomb κυκλοφόρησαν στις 22 Φεβρουαρίου 2011 με επίπεδα API 11- 13. Σχεδιάστηκε για συσκευές με μεγαλύτερες οθόνες, ειδικότερα για τις ταμπλέτες (tablet). [31]
 - Οι εκδόσεις 4.0- 4.0.4 με κωδικό όνομα Ice Cream Sandwich κυκλοφόρησε στις 18 Οκτωβρίου 2011 με επίπεδα API 14-15 προσθέτοντας μια ανανεωμένη αρχική οθόνη στο λειτουργικό σύστημα. [32]
 - Οι εκδόσεις 4.1- 4.3.1 με κωδικό όνομα Jelly Bean κυκλοφόρησαν στις 9 Ιουλίου 2012 με επίπεδα API 16-18. Οι εκδόσεις αυτές εστιάστηκαν στην βελτίωση της απόδοσης και το σχεδίασαν ώστε να δώσει στο λειτουργικό μια ομαλότερη και πιο δεκτική αίσθηση. [33]
 - Οι εκδόσεις 4.4- 4.4.4 με κωδικό όνομα KitKat κυκλοφόρησαν στις 31 Οκτωβρίου 2013 με επίπεδο API 19-20. το KitKat επικεντρώθηκε κυρίως στη βελτιστοποίηση του

λειτουργικού συστήματος και τη βελτίωση των επιδόσεων σε συσκευές entry-level που διαθέτουν περιορισμένους πόρους. [34]

- Οι εκδόσεις 5.0- 5.1.1 με κωδικό όνομα Lollipop κυκλοφόρησαν στις 12 Νοεμβρίου 2014 με επίπεδο API 21-22. Μία από τις πιο βασικές αλλαγές στην κυκλοφορία του Lollipop είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μια σχεδιαστική γλώσσα γνωστή ως Material Design, η οποία έγινε για να διατηρήσει μια αίσθηση χαρτιού στο περιβάλλον. [35]

- Οι εκδόσεις 7.0- 7.1.1 με κωδικό όνομα Marshmallow κυκλοφόρησαν στις 5 Οκτωβρίου 2015 με επίπεδο API 23. Το Marshmallow επικεντρώθηκε κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη. [36]

- Οι εκδόσεις 7.0- 7.1.1 με κωδικό όνομα Nougat κυκλοφόρησε στις 22 Αυγούστου 2016 με επίπεδο API 24. Το Nougat εισάγει σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του, συμπεριλαμβανομένης της δυνατότητας να εμφανίσει πολλαπλές εφαρμογές στην οθόνη με προβολή διαίρεσης της οθόνης. [37]

- 21 Αυγούστου 2017 κυκλοφόρησε η όγδοη μεγαλύτερη έκδοση του κινητού λειτουργικού Android με κωδικό όνομα Oreo ή O. Τα κυριότερα χαρακτηριστικά του είναι ότι οι ειδοποιήσεις μπορούν να αναβληθούν και να μουν σε ομάδες που βασίζονται με θέματα γνωστές ως κανάλια, καθώς επίσης περιέχει ολοκληρωμένη υποστήριξη για τις λειτουργίες εικόνας σε εικόνα που υποστηρίζεται από την εφαρμογή YouTube. [38]

2.5 IOS



Εικόνα 14 iOS

Το iOS είναι ένα λογισμικό για κινητά τηλέφωνα που δημιουργήθηκε από την Apple Inc. και παρουσιάστηκε στα κινητά τηλέφωνα iPhone το 2007. Η εταιρεία Apple (μέχρι τις 9 Ιανουαρίου του 2007 ονομαζόταν Apple Computer Inc.) γνωστή και ως Apple είναι μια αμερικανική πολυεθνική εταιρεία τεχνολογίας με έδρα το Κουπερτίνο της Καλιφόρνια. Η Apple ιδρύθηκε από τον Steve Jobs και τον Steven Wozniak το 1976 σε ένα γκαράζ

στην μικρή πόλη Λος Άλτος της Καλιφόρνια. Το 2016 δήλωσε κέρδη πάνω από 215 δις δολάρια και απασχολούσε περισσότερους από 116.000 εργαζόμενους. Η δραστηριότητα της είναι ο σχεδιασμός, η ανάπτυξη και η πώληση καταναλωτικών ηλεκτρονικών ειδών, λογισμικού, υπολογιστών και κινητών τηλεφώνων. Τα πιο γνωστά προϊόντα της είναι οι υπολογιστές της σειράς Macintosh, το λειτουργικό σύστημα Mac OS X, το οποίο είναι μια σειρά γραφικών λειτουργικών συστημάτων που δημιουργήθηκε από την Apple Inc και συμπεριλαμβάνεται σε κάθε υπολογιστή Macintosh, η φορητή συσκευή αναπαραγωγής πολυμέσων iPod, το έξυπνό τηλέφωνο iPhone, και ο υπολογιστής ταμπλέτα iPad.

Το 1985 το διοικητικό συμβούλιο της Apple αποφάσισε να 'εξορίσει' τον συνιδρυτή της εταιρείας Steve Jobs, ο οποίος και αποχώρησε από αυτή και ίδρυσε την εταιρεία NeXT Computer Inc. Μεταξύ 1985 και 1992 η Apple κατείχε κυρίαρχη θέση στην αγορά προσωπικών υπολογιστών με γραφικό περιβάλλον. Οι υπολογιστές Macintosh αποτελούσαν μηχανήματα υψηλής ποιότητας και τιμής, και ιδιαίτερης ευκολίας στη χρήση. Το 1992 η Apple μεταπήδησε στη πλατφόρμα επεξεργαστών PowerPC σε συνεργασία με τις εταιρείες Motorola και IBM. Η συνεχιζόμενη ανάπτυξη λειτουργικού συστήματος Windows από την Microsoft, καθώς και τα εσωτερικά προβλήματα της εταιρείας, τόσο τεχνολογικά και λειτουργικά (παρωχημένο λειτουργικό σύστημα, έλλειψη στρατηγικών στόχων, αυξημένος ανταγωνισμός και απώλεια εσόδων) όσο και διοικητικά- πολιτικά (διαφωνίες μεταξύ

στελεχών, αδυναμία λήψης βασικών αποφάσεων) έφεραν την Apple στα πρόθυρα πτώχευσης το 1996.

Μετά από αρκετές διαπραγματεύσεις σχετικά με την αγορά λογισμικού ως το νέο λειτουργικό σύστημα της εταιρείας, η Apple αγόρασε την NeXT τον Φεβρουάριο του 1997. Λίγο αργότερο ο Steve Jobs επέστρεψε στην εταιρεία ως προσωρινός Διευθύνων Σύμβουλος και αργότερα μονιμοποιήθηκε. [39]

2.6 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ IOS

- Πρώτη έκδοση του λειτουργικού συστήματος iPhone ήταν το iPhone OS 1. Κυκλοφόρησε στις 6 Μαρτίου 2008.

- Δεύτερη έκδοση του λειτουργικού συστήματος iPhone ήταν το iPhone OS 2. Κυκλοφόρησε στις 11 Ιουλίου 2008. Το iPhone 3G ήταν το πρώτο κινητό της εταιρείας που είχε την εφαρμογή App Store για την εγκατάσταση εφαρμογών.

- Το iPhone OS 3 έγινε διαθέσιμο στους χρήστες στις 14 Ιουνίου 2009 και ήταν εγκατεστημένο στο κινητό τηλέφωνο iPhone 3GS.

- Το iOS 4 έγινε διαθέσιμο στις 21 Ιουνίου 2010. Το λογισμικό αυτό είχε την δυνατότητα να εγκατασταθεί στα κινητά της εταιρείας που είχαν προηγούμενη έκδοση του iPhone.

- Το iOS 5 κυκλοφόρησε στις 6 Ιουνίου μαζί με το iPhone 4S. Κύρια αλλαγή ήταν η προσθήκη του κέντρου ειδοποιήσεων.

- Το iOS 6 κυκλοφόρησε στις 11 Ιουνίου 2012. Το συγκεκριμένο λογισμικό δεν υποστηριζόταν από όλα τα κινητά της εταιρείας που είχαν κυκλοφορήσει νωρίτερα διότι είχε αρκετά υψηλές απαιτήσεις. Εκείνο το διάστημα έγινε η παρουσίαση του iPhone 5.

- Το iOS 7 κυκλοφόρησε στις 10 Ιουνίου 2013 με την έκδοση του iPhone 5C και 5S, είναι η τελευταία αναβάθμιση που μπορούσε να γίνει στο iPhone 4.

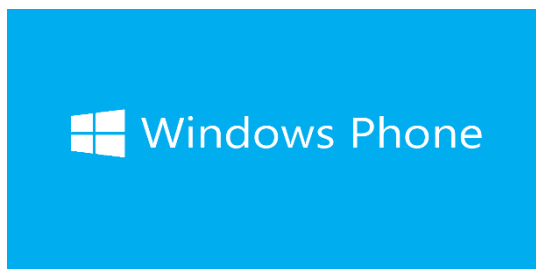
- Το iOS 8 κυκλοφόρησε στις 2 Ιουνίου 2014 στην ετήσια έκθεση της Apple World Wide Developers Conference.

- Το iOS 9 κυκλοφόρησε στις 8 Ιουνίου 2015, και είναι συμβατό με όλες τις συσκευές από το iPhone 4S μέχρι το iPhone 6S plus.

- Το iOS 10 ανακοινώθηκε στις 13 Ιουνίου 2016. Ήταν η αρχική έκδοση για το iPhone 7 και το iPhone 7 plus, και για αυτό το λόγο η παρουσίαση του έγινε στο iPhone 7.

- Το iOS 11 ανακοινώθηκε στις 5 Ιουνίου 2017 και το πρώτο κινητό που θα κυκλοφορήσει με προεγκατεστημένο το iOS 11 θα είναι το iPhone 8 ή iPhone X που θα κυκλοφορήσει 12 Σεπτεμβρίου 2017. [40]

2.7 WINDOWS PHONE



Εικόνα 15 Windows Phone

Όπως αναφέρθηκε παραπάνω, ένα από τα τρία λειτουργικά για κινητά τηλέφωνα (smartphones) είναι και το Windows, της εταιρείας Microsoft. Η Microsoft είναι μια Αμερικάνικη εταιρία η οποία ιδρύθηκε το 1975 από τον Bill Gates και τον Paul Allen. Ένα από τα πιο σημαντικά προϊόντα της εταιρείας είναι το λειτουργικό Windows που καταλαμβάνει περίπου το 90% του ποσοστού της αγοράς για ηλεκτρονικούς υπολογιστές. Η

Microsoft είναι από τις μεγαλύτερες εταιρίες απασχολώντας περισσότερους από 114.000 εργαζόμενους. Τα Windows Phone αναπτύχθηκαν το 2010, και το λειτουργικό αυτό έχουν κάποια μοντέλα της Nokia, Samsung, HTC, Huawei και ZTE.

Οι εκδόσεις των Windows ξεκίνησαν με τα πρώτα Windows mobile, τα οποία κυκλοφόρησαν στην αγορά το 1996, με το λειτουργικό τους βασισμένο σε αυτό του σταθερού υπολογιστή. Τα κινητά αυτά λόγω του συγκεκριμένου λειτουργικού συστήματος, μπορούσαν να έχουν μεγάλες οθόνες, αλλά μικρές δυνατότητες. Επίσης, πρόσφεραν την δυνατότητα αναβάθμισης. Αυτό έκανε πολλούς να στραφούν προς τα Windows mobile γιατί προσέφεραν ευκολία σύνδεσης με τους server των επιχειρήσεων (όπως η Microsoft SQL server). Στην συνέχεια, λόγω μεγάλου ανταγωνισμού, έλλειψη εφαρμογών και δυνατοτήτων βγήκαν εκτός ανταγωνισμού το 2010. Η κυκλοφορία των windows mobile σταμάτησε και κυκλοφόρησαν στην αγορά τα Windows Phone 7, τα οποία ήταν άμεσα συνδεδεμένα με τις υπηρεσίες της Microsoft.

Στο επόμενο κεφάλαιο θα δούμε την ιστορική αναδρομή των εκδόσεων Windows Phone και τις διαφορές μεταξύ τους.

2.8 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ WINDOWS PHONE

- Το Windows Phone 7 κυκλοφόρησε δημόσια στις 8 Νοεμβρίου 2010 στις Ηνωμένες Πολιτείες και ήταν άμεσα συνδεδεμένο με τις υπηρεσίες της Microsoft.

- Το Windows Phone 7.5 κυκλοφόρησε το 2011. Κάποιες από αυτές τις συσκευές βγήκαν με μικρότερο επεξεργαστή και μνήμη RAM, λόγω των μειωμένων απαιτήσεων του λειτουργικού συστήματος.

- Το Windows Phone 7.8 κυκλοφόρησε στις 29 Οκτωβρίου 2012 και είναι η αναβαθμισμένη έκδοση του 7.5. Έχει τα διπλάσια χρώματα από το προηγούμενο λειτουργικό, καινούρια έναρξη και άλλα χαρακτηριστικά που το κάνουν πιο εύχρηστο και φιλικό για το χρήστη.

- 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. Κυριότερη αλλαγή του ήταν η προσθήκη της μπάρας ενημερώσεων.

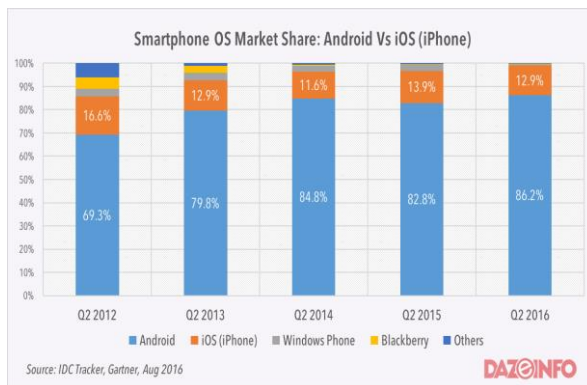
- Το Windows Phone 10 κυκλοφόρησε στις 21 Ιανουαρίου 2015 και η κύρια αλλαγή του ήταν η ενοποίηση του με το Windows 10 για υπολογιστές.

2.9 ΣΥΓΚΡΙΣΗ ΜΕΤΑΞΥ ΤΩΝ ΛΕΙΤΟΥΡΓΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΓΙΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ



Εικόνα 16 Διαμάχη Λογισμικών Κινητών Τηλεφώνων

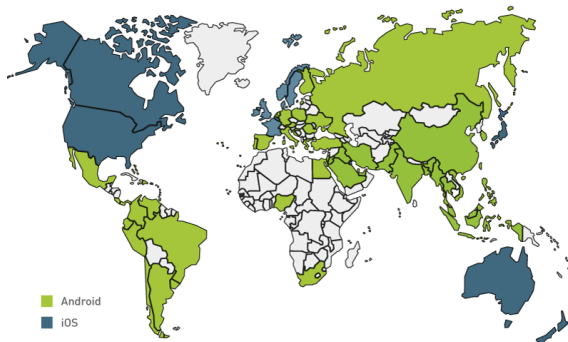
Ανάμεσα στις εταιρίες που έχουν αναπτύξει τα προαναφερθέντα λειτουργικά υπάρχει συνεχής ανταγωνισμός. Ο ανταγωνισμός έχει σαν αποτέλεσμα την βελτίωση των προϊόντων που φτάνουν στους χρήστες. Όπως στην πλειοψηφία προϊόντων που παράγονται από διαφορετικές εταιρίες, υπάρχει και στο καταναλωτικό κοινό των κινητών τηλεφώνων μια διαμάχη σχετικά με τις προτιμήσεις τους. Σε αυτό το κεφάλαιο θα παρουσιάσω τα ποσοστά προτίμησης του κοινού, μια αντικειμενική σύγκριση των λειτουργικών συστημάτων, καθώς και τα πλεονεκτήματα και μειονεκτήματά τους.



Εικόνα 17 Ποσοστά Προτίμησης Λογισμικού

αυξάνεται σταδιακά από το 2012 και έπειτα, ενώ το ποσοστό των χρηστών που προτίμησε κινητές συσκευές με λειτουργικό σύστημα iOS έχει μικρές διακυμάνσεις. [42]

Σύμφωνα με το Moon Technolabs Pvt Ltd (εταιρία ανάπτυξης εφαρμογών για κινητά iPhone και Android), παρατηρούμε ότι για το 2016 το 87% των καταναλωτών προτίμησε συσκευές με λειτουργικό σύστημα Android, ενώ μόλις το 12% προτίμησε συσκευές με λειτουργικό σύστημα iOS. Το αντίστοιχο ποσοστό των κινητών με λειτουργικό σύστημα Windows Phone είναι μικρότερο από 1%. Επιπλέον, από το ίδιο διάγραμμα παρατηρούμε ότι το ποσοστό των χρηστών που προτίμησε συσκευές με λειτουργικό σύστημα Android,



Εικόνα 18 Προτίμηση Λογισμικού ανά Χώρα

Παρατηρούμε επίσης, ότι το Android καταλαμβάνει το μεγαλύτερο μερίδιο της αγοράς σε αναπτυσσόμενες χώρες όπως Ασία, Αφρική, Νότια Αμερική και κάποιες χώρες τις Ευρώπης, ενώ το iOS έχει μεγαλύτερα ποσοστά στις ανεπτυγμένες χώρες όπως οι ΗΠΑ, η Αυστραλία και τις υπόλοιπες χώρες της Ευρώπης (Γαλλία, Ηνωμένο Βασίλειο και Σκανδιναβικές χώρες).

Το λειτουργικό σύστημα iOS της Apple που διατίθεται μόνο στις συσκευές της εταιρίας είναι ένα πολύ αξιόπιστο λειτουργικό, ικανό να αντεπεξέλθει και στην πιο απαιτητική χρήση, προσφέροντας μεγάλο ποσοστό ασφάλειας δεδομένων. Σε αυτό συμβάλει και το γεγονός ότι είναι λογισμικό κλειστού κώδικα. Απ' την άλλη πλευρά, αν και το App Store της Apple περιέχει πολλές εφαρμογές, το μεγαλύτερο ποσοστό τους διατίθεται μόνο με πληρωμή. [43]

Το λειτουργικό σύστημα Android διατίθεται ελεύθερα από την Google, και αυτό έχει σαν αποτέλεσμα να υπάρχει μεγαλύτερη ποικιλία κινητών που το διαθέτουν (πχ Samsung, LG, Huawei, Xiaomi κλπ.). Θεωρείτε εύκολο στην χρήση του αλλά μερικές φορές είναι ασταθές με συνέπεια να χρειάζεται επανεκκίνηση ή αναβάθμιση για να λυθεί το πρόβλημα. Σε αντίθεση με το App Store, στο Play Store, η πλειοψηφία των εφαρμογών διατίθενται δωρεάν και δίνεται η δυνατότητα να αφαιρεθούν οι διαφημίσεις, έναντι μικρού αντιτίμου. Ένα ακόμα ελάττωμα του, είναι ότι λόγω του λογισμικού του που είναι ανοιχτού κώδικα, δεν θεωρείται αρκετά ασφαλές. Ο πηγαίος κώδικας διατίθεται ελεύθερα σε όσους ζητήσουν να τον εξετάσουν, ή ακόμα και να τον τροποποιήσουν. [44]

Τέλος, το λειτουργικό σύστημα Windows διατίθεται από την Microsoft και το συναντάμε σε λιγότερες εταιρίες (Microsoft, Nokia και παλιότερα σε κάποια μοντέλα της Samsung). Αν και πλέον είναι το ίδιο σύστημα με αυτό που χρησιμοποιείται στους υπολογιστές με ιδιαίτερη επιτυχία, παλιότερα υπήρχε ειδική έκδοση για κινητά τηλέφωνα smartphone, η οποία παρουσίαζε αρκετά προβλήματα. Κάποια από αυτά είναι η περιορισμένη ευχρηστιά τους, και το μικρό ποσοστό εφαρμογών που διατίθενται στο Windows Store. Το σημαντικό πλεονέκτημα είναι η δυνατότητα χρήσης δημοφιλή προγραμμάτων που θεωρούνται απαραίτητα στον υπολογιστή (πχ Microsoft Office) και ο συγχρονισμός των συσκευών.

2.10 ΔΙΑΔΙΚΑΣΙΑ ΑΝΕΒΑΣΜΑΤΟΣ ΕΦΑΡΜΟΓΗΣ ΑΝΑ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ

Κάθε λογισμικό έχει ξεχωριστό τρόπο ανεβάσματος εφαρμογών στην δική τους εφαρμογή διανομής εφαρμογών (PLIAS Store, App Store, Windows Store). Ακολουθώντας συγκεκριμένα βήματα, μπορείς κανείς να κάνει μια εφαρμογή διαθέσιμη για τους χρήστες smartphones. Με την συγκεκριμένη κίνηση ο προγραμματιστής μπορεί να κερδίζει χρήματα είτε μέσω διαφημίσεων, είτε διαθέτοντας την εφαρμογή επί πληρωμή.

Play Store



Όταν ολοκληρωθεί η εφαρμογή και θέλεις να την ανεβάσεις στο Play Store, πρέπει:

Εικόνα 19 Play Store

- Να δημιουργηθεί λογαριασμός Developer: πηγαίνουμε στην σελίδα (<https://play.google.com/apps/publish/signup/>) και συμπληρώνουμε τα στοιχεία που απαιτούνται, αποδεχόμαστε τους όρους χρήσης (εφόσον διαβαστούν πρώτα). Για την ολοκλήρωση αυτής της διαδικασίας χρειάζεται να πληρωθεί το αντίτιμο των 20 ευρώ.

- Να προσθέσουμε τα στοιχεία της εφαρμογής: επιλέγουμε Add new application και στην συνέχεια την γλώσσα στην οποία έχουμε δημιουργήσει την εφαρμογή, το όνομα της και στην συνέχεια πατάμε Prepare Store Listing.

- Να δημιουργηθεί η περιγραφή της εφαρμογής, σύντομο κείμενο διαφήμισης και τις αναβαθμίσεις (όταν υλοποιήσουμε). Η περιγραφή πρέπει να είναι αναλυτική και μας δίνεται η δυνατότητα, επιλέγοντας το Add Translations να προσθέσουμε περιγραφή και σε άλλες γλώσσες.

- Να προσθέσουμε εικονίδια και βίντεο: είναι υποχρεωτικό να τοποθετήσουμε τουλάχιστον 2 εικόνες και να ανεβάσουμε βίντεο με την χρήση της εφαρμογής YouTube.

- Επιλέγουμε την κατηγορία: πρέπει να επιλεγθεί αν πρόκειται για εφαρμογή ή παιχνίδι, καθώς και την κατηγορία του κοινού για το οποίο είναι κατάλληλη.

- Ανέβασμα εφαρμογής: Στην αριστερή στήλη της πλατφόρμα επιλέγουμε APK και επιλέγουμε την εφαρμογή (.apk). Στην συνέχεια με βάση τις παραμέτρους που έχουν οριστεί στο AndroidManifest.xml εμφανίζονται οι συσκευές για τις οποίες είναι διαθέσιμη η εφαρμογή. Αν κρίνουμε αναγκαίο, μπορούμε να αποεπιλέξουμε μια ή περισσότερες συσκευές.

- Τρόπος διάθεσης της εφαρμογής: επιλέγουμε αριστερά Pricing And Distribution και επιλέγουμε αν θέλουμε να διατίθεται δωρεάν ή επί πληρωμή και τις χώρες στις οποίες επιθυμούμε να είναι διαθέσιμη (από Ελλάδα μπορούμε να ανεβάσουμε μόνο δωρεάν εφαρμογές).

- Μπαίνουμε στο Play Store και αναζητούμε την εφαρμογή! [45]

App store



Όταν είναι έτοιμη η εφαρμογή για κινητό τηλέφωνο με λογισμικό iOS και θέλουμε να την κάνουμε διαθέσιμη και για άλλους χρήστες μέσω της εφαρμογής διανομής εφαρμογών της Apple (App Store), ακολουθούμε τα εξής βήματα:

Εικόνα 20 App Store

- Συγκεντρώνουμε τις πληροφορίες της εφαρμογής (στιγμιότυπα οθόνης, όνομα, περιγραφή, λέξεις κλειδιά, url υποστήριξης, url μάρκετινγκ, εικονίδιο εφαρμογής, κατηγορία, βαθμολογία βάση ερωτηματολογίου, πνευματική ιδιοκτησία).

- Δημιουργούμε αναγνωριστικό πακέτου (Bundle Identifier). Πηγαίνουμε στην ιστοσελίδα (<https://developer.apple.com/account/overview.action>) και πραγματοποιούμε σύνδεση. Πατάμε στην εντολή Identifiers και μετά στο σύμβολο “+” στην πάνω δεξιά γωνία της οθόνης. Έπειτα, ονομάζουμε το αναγνωριστικό εφαρμογής (Bundle ID) με το ίδιο όνομα με την εφαρμογή μας. Επιλέγουμε τις υπηρεσίες εφαρμογής που χρειάζεται η εφαρμογή για να λειτουργήσει, και κάνουμε κλικ στο κουμπί Continue. Επιβεβαιώνουμε τις υπηρεσίες και επιλέγουμε υποβολή.

- Δημιουργούμε ένα 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”. Εισάγουμε την διεύθυνση ηλεκτρονικού ταχυδρομείου, επιλέγουμε την επιλογή “Saved to Disk” και πατάμε το επόμενο για να αποθηκεύσουμε το Certificate Signing Request για αργότερα.

- Δημιουργούμε ένα πιστοποιητικό παραγωγής App Store (App Store Production Certificate). Πηγαίνουμε στην ιστοσελίδα που επισκεφτήκαμε παραπάνω. Κάνουμε κλικ στην επιλογή Certificates και μετά επιλέγουμε το σύμβολο “+” στην πάνω δεξιά γωνία της οθόνης. Κάνουμε κλικ στην παραγωγή App Store και μετά επιλέγουμε “Continue”. Μεταφορτώνουμε το “Certificate Signing Request”. Κατεβάζουμε το πιστοποιητικό και το εγκαθιστούμε κάνοντας διπλό κλικ.

- Δημιουργούμε προφίλ παράδοσης παραγωγής (Production Provisioning Profile). Πηγαίνουμε πάλι στην ιστοσελίδα και κάνουμε κλικ στο “Provisioning Profiles” και πάνω δεξιά στην οθόνη επιλέγουμε το σύμβολο “+”. Πατάμε το “App Store Distribution option” και μετά πατάμε “Continue”. Επιλέγουμε το αναγνωριστικό εφαρμογής (Bundle ID) και πατάμε “Continue”. Επιλέγουμε το αίτημα υπογραφής πιστοποιητικού (Certificate Signing Request), πατάμε “Continue και Generate”. Κατεβάζουμε το αρχείο και το εγκαθιστούμε στον υπολογιστή κάνοντας διπλό κλικ.

- 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, επιλέγουμε το αναγνωριστικό εφαρμογής (Bundle ID) που φτιάξαμε νωρίτερα και πατάμε “Create” για να δημιουργηθεί η πρώτη έκδοση της εφαρμογής μας.

- 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 (menu – project - archive)” και συνδεόμαστε ως λογαριασμός προγραμματιστή Apple. Υποβάλουμε την εφαρμογή στο App Store και περιμένουμε για την επιβεβαίωση.

- Συμπληρώνουμε τις πληροφορίες την πρώτης έκδοσης. Καταχωρούμε τις πληροφορίες που συγκεντρώσαμε στο πρώτο βήμα, επιλέγουμε την εφαρμογή που θέλουμε να συνδέσουμε με την έκδοση, συμπληρώνουμε τις πληροφορίες τιμολόγησης και κάνουμε αποθήκευση.
- Υποβάλλουμε την έκδοση για αναθεώρηση. Επιλέγουμε τον τρόπο απελευθέρωσης της εφαρμογής (μη αυτόματη απελευθέρωση, αυτόματη απελευθέρωση) και περιμένουμε να εγκριθεί.
- Στην περίπτωση που έχουμε επιλέξει μη αυτόματη απελευθέρωση, 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.
- Για να ανεβάσουμε την εφαρμογή μας πρέπει να διαθέσουμε 99 ευρώ.
- Μπαίνουμε στο App Store και αναζητούμε την εφαρμογή[46]

Windows Store



Προκειμένου να ανεβάσουμε την εφαρμογή μας στο Windows Store και να την κάνουμε διαθέσιμη και για άλλους χρήστες κινητών Windows Phone, χρειάζεται να έχουμε λογαριασμό στο Dev Center και να κάνουμε τα εξής βήματα:

- Εικόνα 21 Windows Store*
- Ανοίγουμε την ιστοσελίδα (<https://dev.windows.com>) και συνδεόμαστε.
 - Στην σελίδα που θα εμφανιστεί (Dashboard) πατάμε στην επιλογή “Create a new app”.
 - Επιλέγουμε όνομα για την εφαρμογή και εφόσον είναι διαθέσιμο (εμφανιστεί πράσινη σήμανση δίπλα στο όνομα της εφαρμογής) πατάμε “Reserve app name”.
 - Πατάμε την επιλογή “Start your submission” για υποβάλουμε τα στοιχεία της εφαρμογής
 - Επιλέγουμε “Pricing and availability” και συμπληρώνουμε το κόστος που θέλουμε να πληρώσει ο χρήστης που θέλει να αποκτήσει την εφαρμογή και αν θα διατίθεται για δοκιμή συγκεκριμένου χρόνου. Επιλέγουμε “Publish this app soon as it passes certification” και “save”.
 - Επιλέγουμε “App properties” για να δηλώσουμε σε ποια κατηγορία ανήκει η εφαρμογή (πχ παιχνίδι, εκπαίδευση) και την ηλικία για την οποία είναι διαθέσιμη η εφαρμογή (3+, 7+, 12+, 16+, 18+) και “Save”.
 - Φορτώνουμε την εφαρμογή. Πατάμε το browse your files και επιλέγουμε την θέση της εφαρμογής. Πατάμε “Save”.
 - Όταν ολοκληρωθεί η μεταφόρτωση του πακέτου, κάνουμε κλικ στην επιλογή “Save”.
 - Δίνουμε περιγραφή εφαρμογής και εισάγουμε την λειτουργία της εφαρμογής και επιπλέον στοιχεία που απαιτούνται. Κάνουμε κλικ στην επιλογή “Save”.
 - Ελέγχουμε τις λεπτομέρειες και πατάμε στην επιλογή “Submit to the Store”.
 - Το κόστος ανεβάσματος εφαρμογής στο Windows Store είναι περίπου 20 ευρώ
 - Μπαίνουμε στο Windows Store και αναζητούμε την εφαρμογή! [47]

3 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΣΕ ANDROID

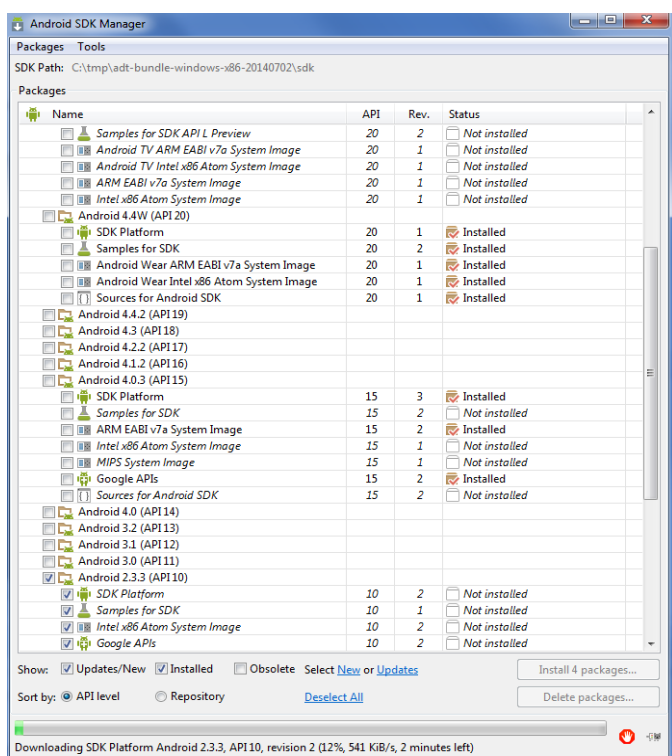
Στο κεφάλαιο αυτό θα αναφερθούμε στον τρόπο δημιουργίας εφαρμογών Android. Με ποιον τρόπο αναπτύσσονται εφαρμογές στο παρελθόν, πόσο εξελίχθηκε ο τρόπος ανάπτυξης και πως αναπτύσσονται στις τωρινές εποχές. Θα αναλυθεί ο τρόπος εγκατάστασης του Android Plugin στο περιβάλλον του eclipse (το περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής που θα αναλυθεί στο επόμενο κεφάλαιο). Επίσης θα αναφερθεί η χρήση του Android για πολυμέσα και ο τρόπος που αποθηκεύονται τα δεδομένα σε κινητά τηλέφωνα με λειτουργικό σύστημα Android.

3.1 ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ANDROID PLUGIN ΣΕ ECLIPSE

Όταν ολοκληρωθεί επιτυχώς η εγκατάσταση του eclipse και του Android Development Tools (ADT), θα πρέπει να κάνουμε εγκατάσταση τις εκδόσεις του Android που θα χρησιμοποιήσουμε και να προβούμε στις κατάλληλες ρυθμίσεις του Android Virtual Device Manager προκειμένου να δημιουργήσουμε μια εικονική συσκευή στην οποία θα βλέπουμε τα αποτελέσματα που δημιουργούνται από τον κώδικα που έχουμε γράψει.

Android SDK Manager

Επιλέγοντας το Android SDK Manager, μας δίνεται η δυνατότητα να επιλέξουμε την έκδοση του Android που θα χρησιμοποιήσουμε ξεκινώντας από την έκδοση Android 1.5 (API 3) έως την έκδοση Android 4.0.3 (API 15) (όπως φαίνεται στην Εικόνα 22). Σημαντικό είναι ότι για κάποιες εκδόσεις, η εγκατάσταση του ARM EABI v7a System Image είναι απαραίτητη προϋπόθεση για να τρέξει το Android Virtual Device Manager. [48]



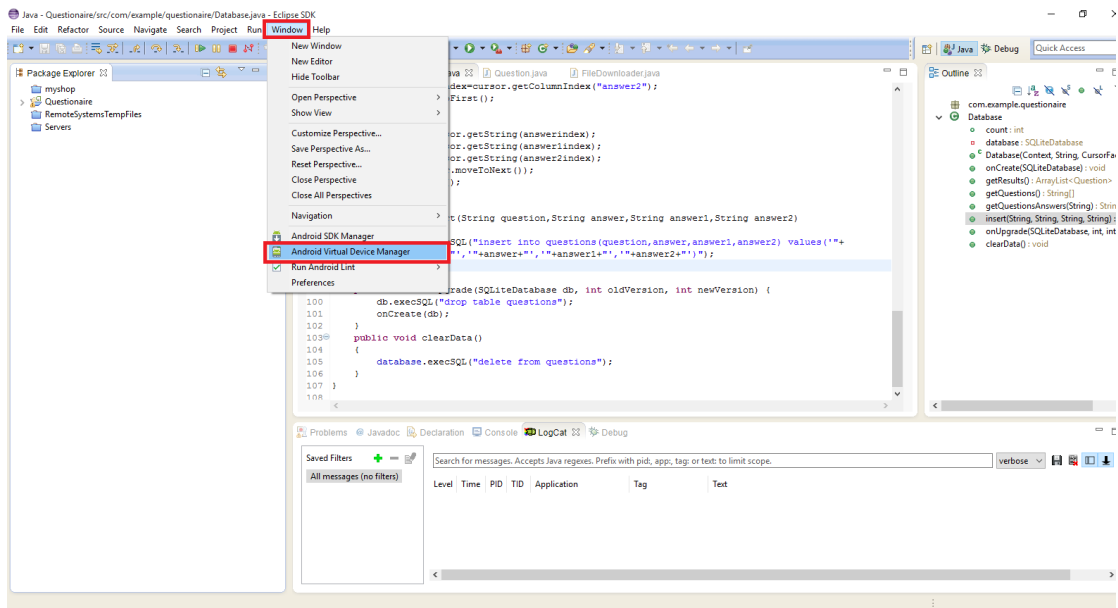
Εικόνα 22 Android SDK Manager

Android Virtual Manager

Με την χρήση του Android Virtual Manager έχουμε την δυνατότητα να δημιουργήσουμε εικονικές συσκευές (smartphone, tablet) και να δούμε την εφαρμογή μας, να λειτουργεί.

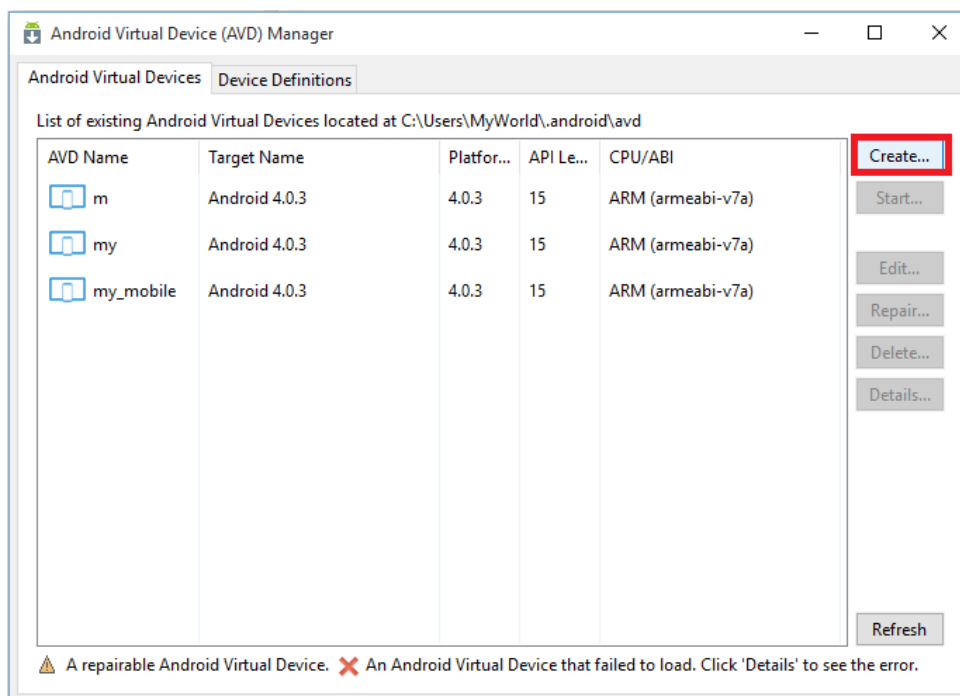
Για να δημιουργήσουμε μια εικονική συσκευή πρέπει:

Να Πατήσουμε Android Virtual Device Manager στην επιλογή Window (όπως φαίνεται στην Εικόνα 23). [49]



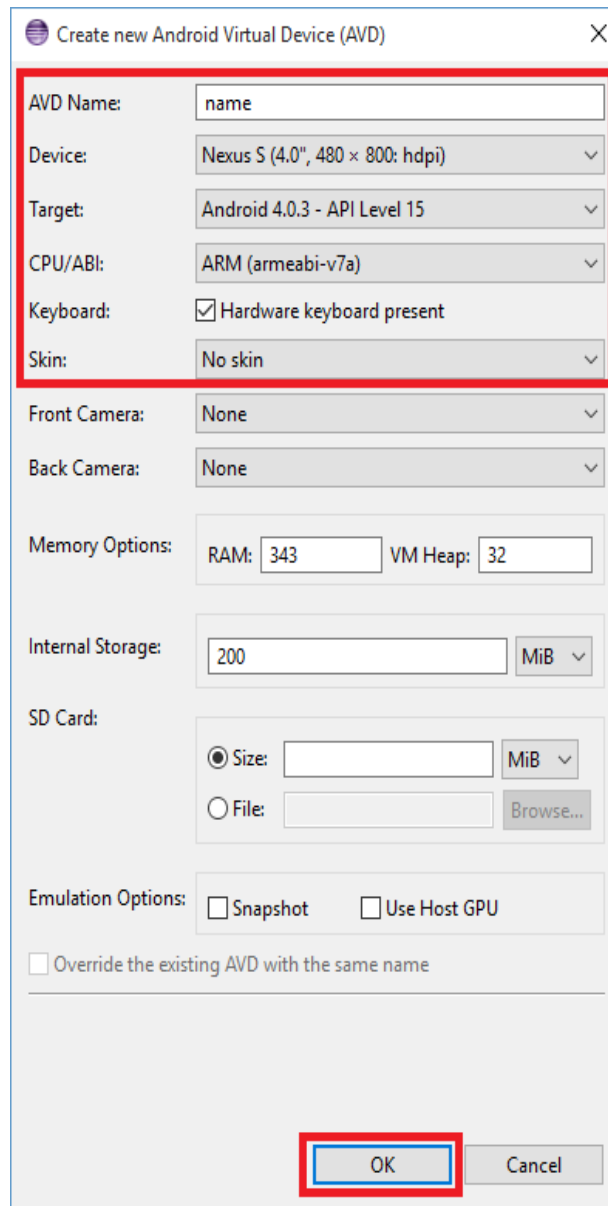
Εικόνα 23 Open Android Virtual Manager

Στο παράθυρο που θα εμφανιστεί θα επιλέξουμε Create (όπως φαίνεται στην Εικόνα 24).



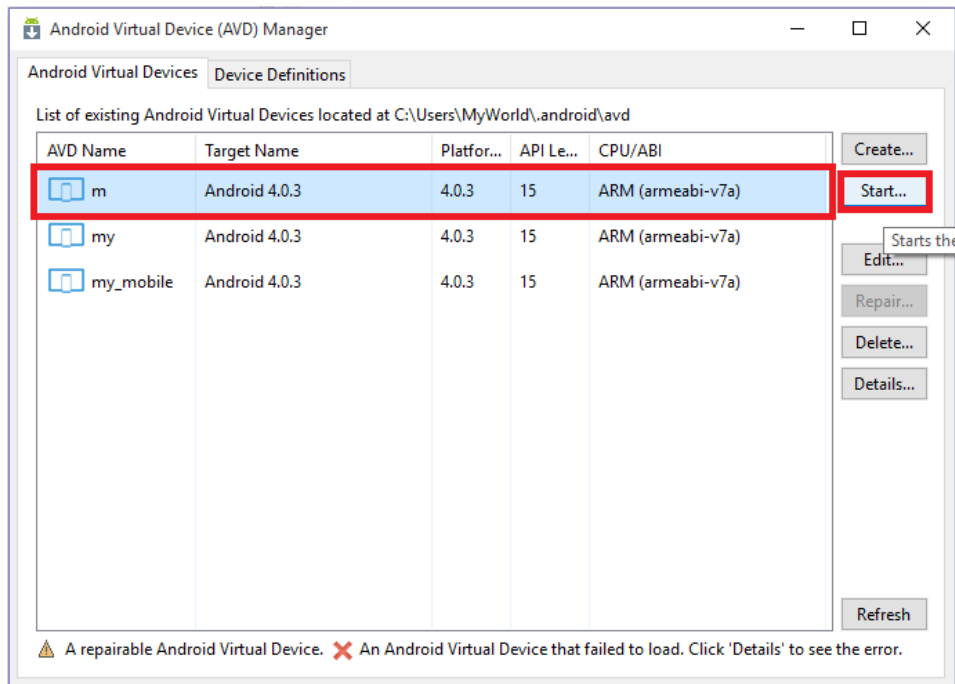
Εικόνα 24 Create Virtual Manager

Συμπληρώνουμε τα στοιχεία που απαιτούνται (όνομα, κινητή συσκευή της επιλογής μας, Android 4.03 – API 15, ARM (armeabi-v7a)) και πατάμε ok. (όπως φαίνεται στην Εικόνα 25).



Εικόνα 25 Create Virtual Manager Informations

Προκειμένου να ανοίξουμε την συσκευή αυτή πρέπει να την επιλέξουμε και να πατήσουμε start (όπως φαίνεται στην Εικόνα 26).



Εικόνα 26 Start Virtual Manager

3.2 ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ANDROID ΠΡΟΓΡΑΜΜΑΤΑ

Για την αποθήκευση δεδομένων σε εφαρμογές με λειτουργικό σύστημα android, δίνεται η δυνατότητα χρήσης αρχείων και βάσεων δεδομένων.

Αρχεία

Json: τύπος αρχείου με συγκεκριμένη μορφή, η οποία χρησιμοποιείται κατά κόρων σε εφαρμογές διαδικτύου. Τα αρχεία αυτά είναι εφικτό να αποθηκευτούν στην εφαρμογή android και να διαχειρίζεται τα δεδομένα που περιέχει γρήγορα και εύκολα.

Xml: τύπος αρχείων που χρησιμοποιούταν συχνότερα στο παρελθόν. Παρ' όλα αυτά, στις περισσότερες διαδικτυακές σελίδες υπάρχει πληθώρα τέτοιων αρχείων. Λόγω της δύσκολης επεξεργασίας τους, υπάρχουν εργαλεία που κάνουν πιο εύκολη την διαχείριση τους. Συνήθως χρησιμοποιούνται για την μορφοποίηση εφαρμογών.

Txt: τύπος αρχείου ο οποίος χρησιμοποιείται συχνότερα για την αποθήκευση κειμένου και όχι τόσο συχνά για την αποθήκευση δεδομένων. Όμως, αν δημιουργηθεί ένα τέτοιο αρχείο ακολουθώντας κάποιους κανόνες, δίνει την δυνατότητα να αποθηκεύσει δεδομένα. Κύριο πρόβλημα είναι η δύσκολη απορρόφηση των δεδομένων επειδή δεν υπάρχουν εργαλεία διαχείρισης τους.

Βάσεις Δεδομένων

SQLite: Οι εφαρμογές που δημιουργούνται για λειτουργικό σύστημα android μπορούν να έχουν αντίστοιχες δυνατότητες με τις βάσεις δεδομένων sql. Δηλαδή μπορούν να

αποθηκευτούν δεδομένα και να επεξεργαστούν με αντίστοιχο τρόπο (πχ. Insert: για την εισαγωγή στοιχείων, delete: για την διαγραφή, select: για την εμφάνιση). Τα δεδομένα αυτά διαγράφονται από το κινητό όταν ο χρήστης επιλέξει εκκαθάριση δεδομένων επιλέγοντας την συγκεκριμένη εφαρμογή στο μενού εφαρμογές των ρυθμίσεων του κινητού. [50]

Στην Εφαρμογή:

Στην εφαρμογή που αναλύεται στην συγκεκριμένη εργασία χρησιμοποιήθηκαν:

Txt: για το κατέβασμα των ερωτήσεων από την ιστοσελίδα (<https://bill09adam01.000webhostapp.com/>). Για την προσπέλαση των δεδομένων, δημιουργήθηκε κώδικας που διαχειριζόταν και αποθήκευε τα δεδομένα στην βάση δεδομένων της εφαρμογής.

SQLite: Βάση δεδομένων της εφαρμογής για την αποθήκευση των ερωτήσεων και των απαντήσεων. Με την χρήση συναρτήσεων, η εφαρμογή διαχειρίζεται τα δεδομένα.

4 Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΕΦΑΡΜΟΓΗ

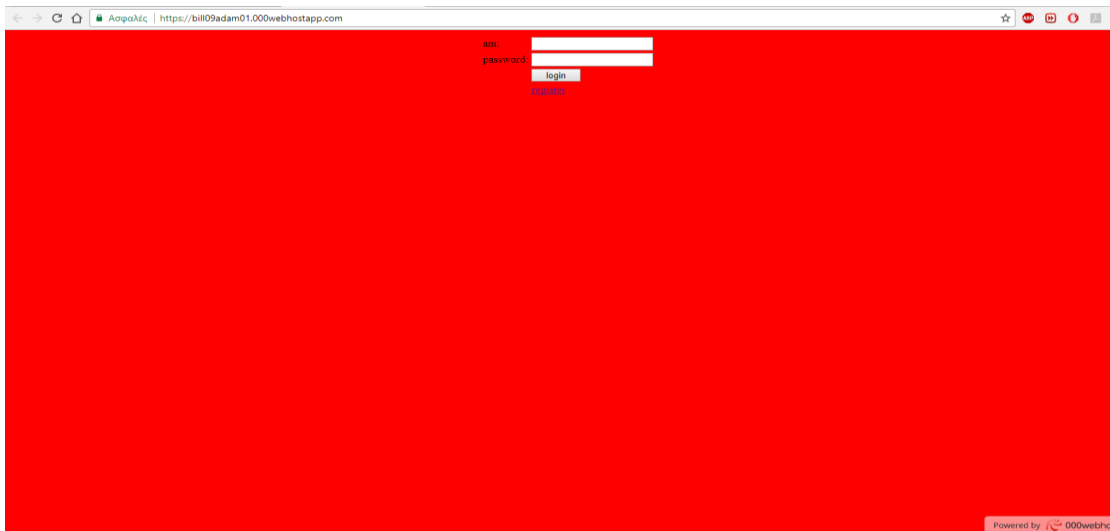
Στο κεφάλαιο αυτό, θα αναλυθεί ο τρόπος λειτουργίας της εφαρμογής. Θα γίνει παρουσίαση του τρόπου δημιουργίας νέου χρήστη στην πλατφόρμα, θα δοθούν παραδείγματα ερωτηματολογίων και τέλος θα γίνει μια επίδειξη του τρόπου που κατεβαίνουν τα ερωτηματολόγια, της σύνδεσης στην εφαρμογή και της διαδικασίας απάντησης του εκάστοτε ερωτηματολογίου.

4.1 ΔΗΜΙΟΥΡΓΙΑ ΧΡΗΣΤΗ

Προκειμένου να καταστεί εφικτή η απάντηση των ερωτηματολογίων από τον χρήστη πρέπει να έχει λογαριασμό για την συγκεκριμένη πλατφόρμα. Είναι αναγκαίο να φτιάξουμε λογαριασμό την πρώτη φορά που θα θελήσουμε να απαντήσουμε ένα ερωτηματολόγιο. Ο λογαριασμός αυτός μπορεί να χρησιμοποιηθεί είτε για την σύνδεση στην ιστοσελίδα (<https://bill09adam01.000webhostapp.com/>), είτε στην απάντηση ερωτηματολογίων. Στην περίπτωση που ο χρήστης θελήσει να συνδεθεί στην εφαρμογή από άλλη συσκευή θα του ζητηθεί να δώσει στοιχεία σύνδεσης.

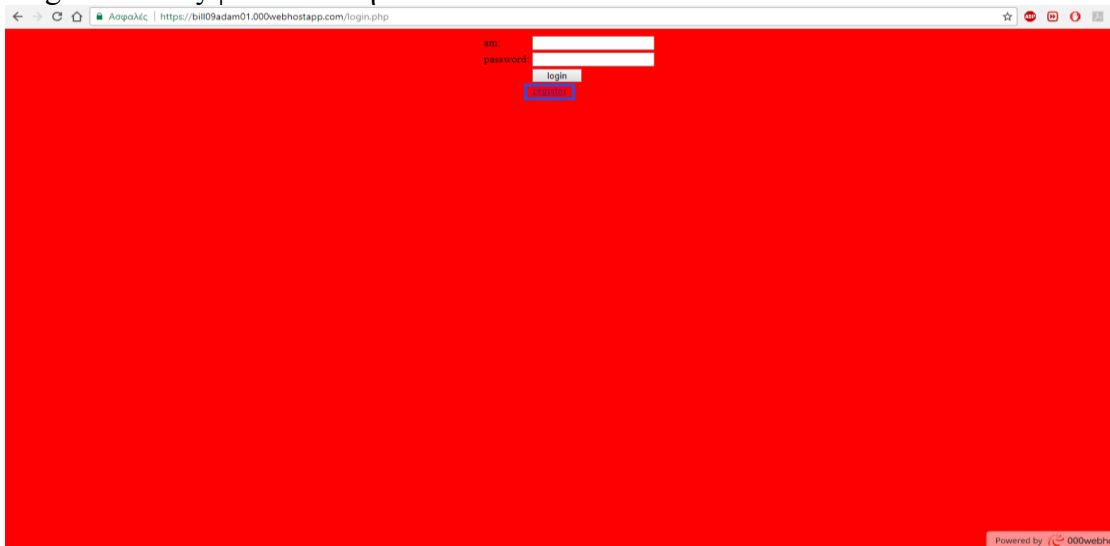
Για να δημιουργηθεί νέος χρήστης πρέπει να συνδεθούμε με την βοήθεια κάποιου φυλλομετρητή (web browser), στην ιστοσελίδα που αναγράφεται παραπάνω.

Εμφανίζεται η εξής σελίδα:



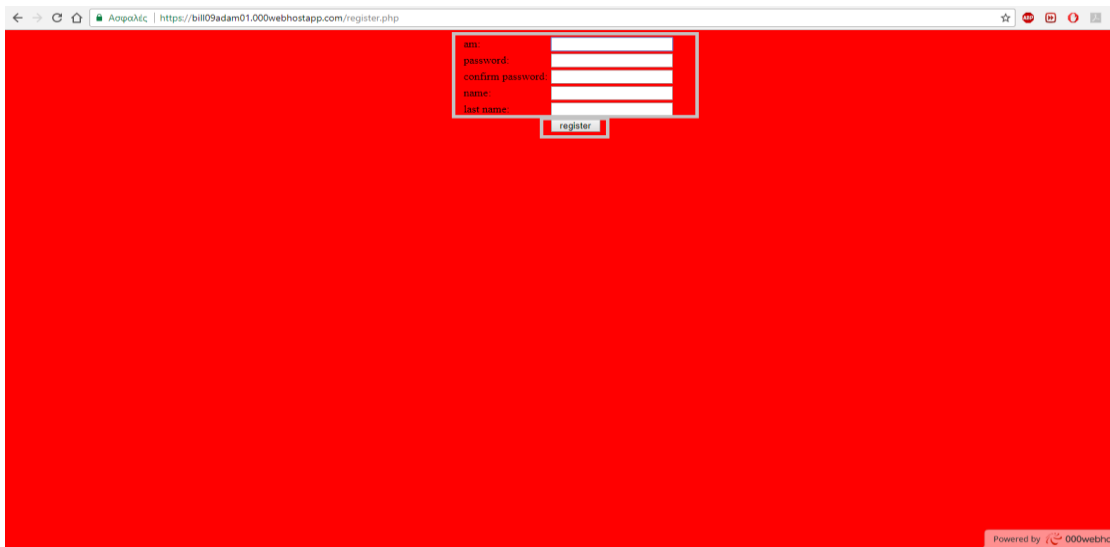
Εικόνα 27 Login and Register to Platform

Για να δημιουργήσει νέο λογαριασμό ο χρήστης χρειάζεται να επιλέξει την επιλογή “register” όπως φαίνεται στην Εικόνα 28.



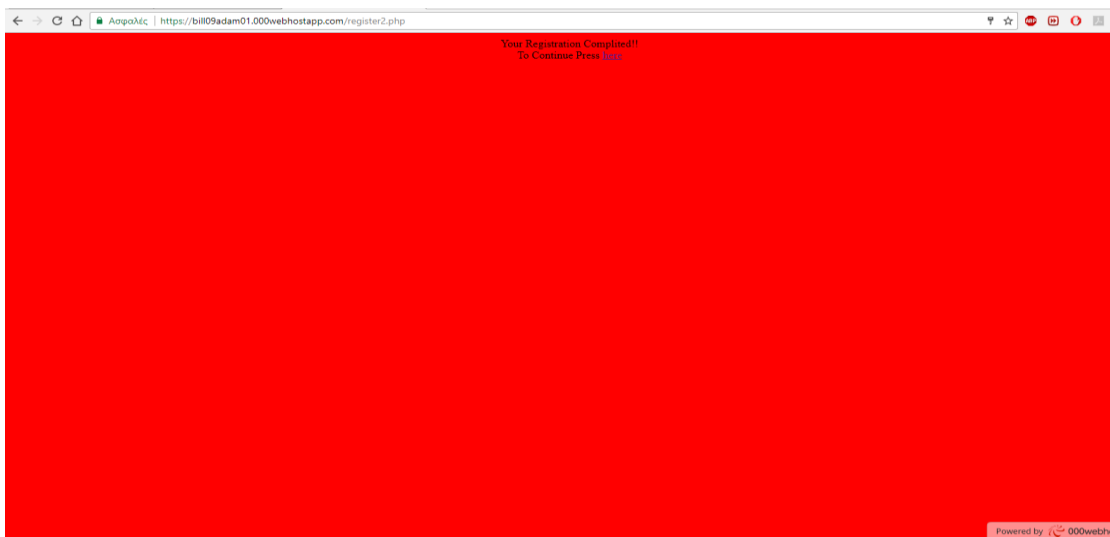
Εικόνα 28 Register to Platform

Τότε θα εμφανιστεί μια φόρμα εγγραφής στην οποία θα πρέπει να συμπληρωθούν τα στοιχεία που ζητούνται :



Εικόνα 29 Fill out the Form to Register

Έπειτα, θα εμφανιστεί μια σελίδα που θα σε ενημερώνει αν έχει ολοκληρωθεί σωστά η διαδικασία εγγραφής.



Εικόνα 30 Registration Completed

Ή ότι υπάρχει κάποιο πρόβλημα (πχ δεν έχουν συμπληρωθεί τα πεδία am, password):



Εικόνα 31 Data Fail

Στην περίπτωση που υπάρχει ήδη χρήστης με αυτά τα στοιχεία:

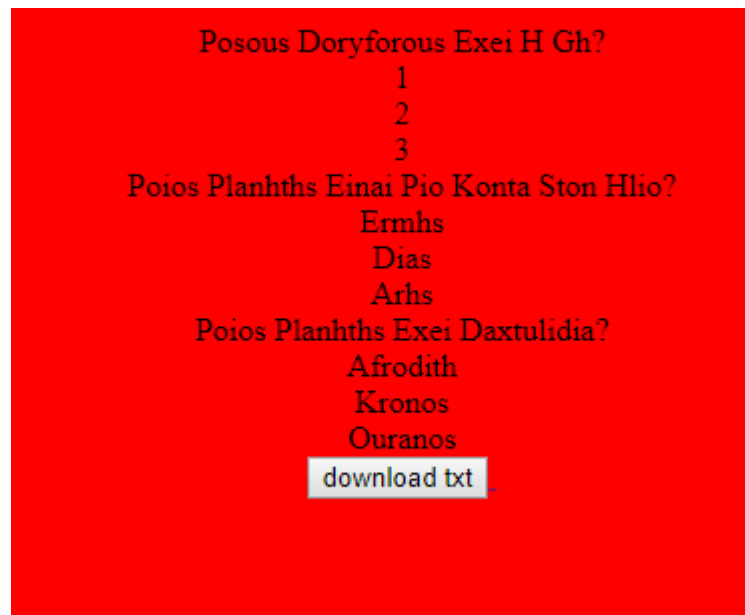


Εικόνα 32 Register Fail

4.2 ΠΑΡΑΔΕΙΓΜΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ

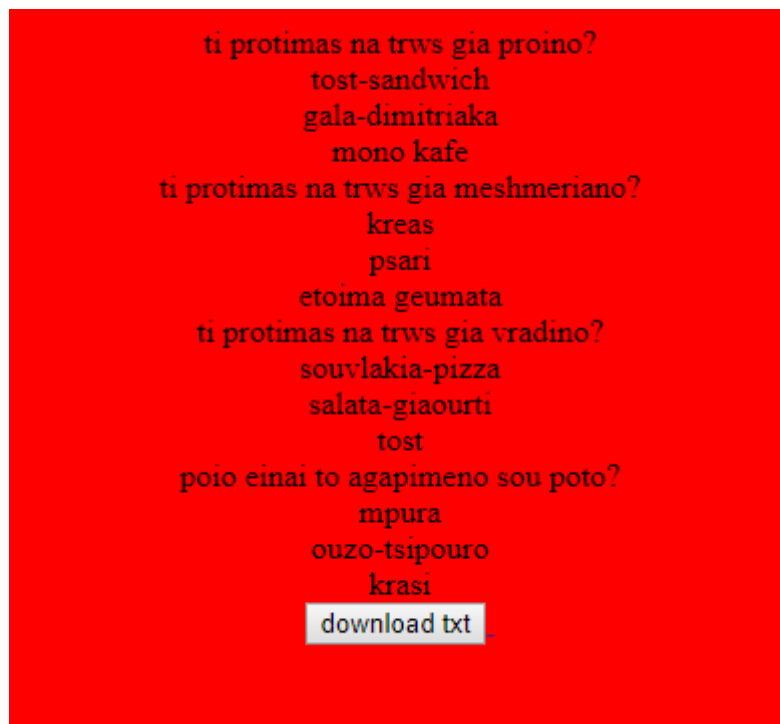
Τα ερωτηματολόγια τα οποία είναι διαθέσιμα προς απάντηση για τους χρήστες τις εφαρμογής είναι (JOY, diatrofi, taksidia και mobile). Δίνεται η δυνατότητα να προστεθούν επιπλέον ερωτηματολόγια ή να αφαιρεθούν αν ο διαχειριστής της πλατφόρμας το θεωρήσει αναγκαίο. Οι ερωτήσεις και οι απαντήσεις των διαθέσιμων ερωτηματολογίων είναι:

JOY:



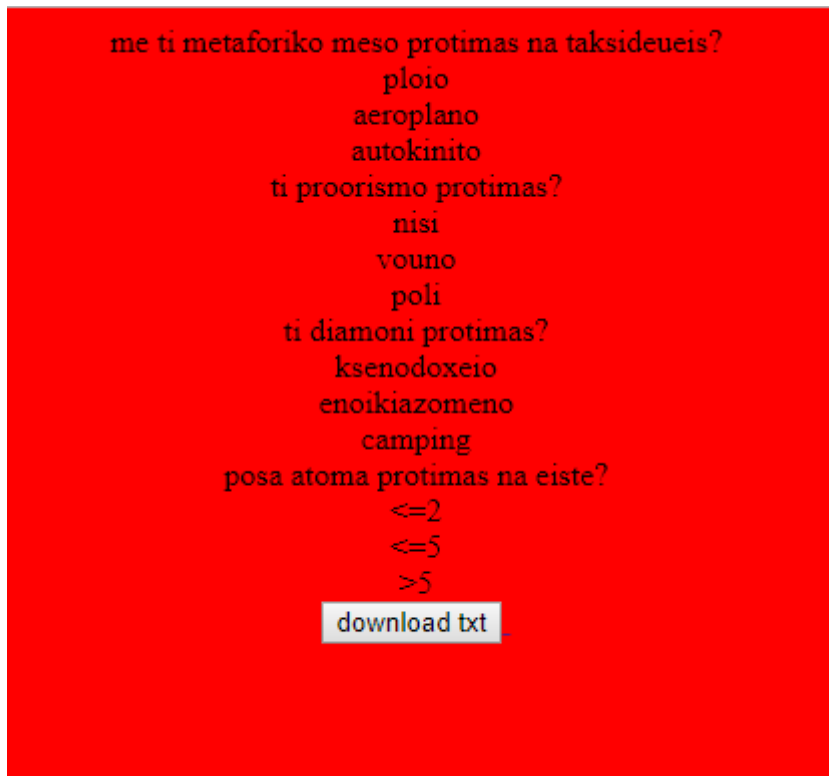
Εικόνα 33 JOY's Questions and Answers

Diatrofi:



Εικόνα 34 Diatrofi's Questions and Answers

Taksidia:



Εικόνα 35 Taksidia's Questions and Answers

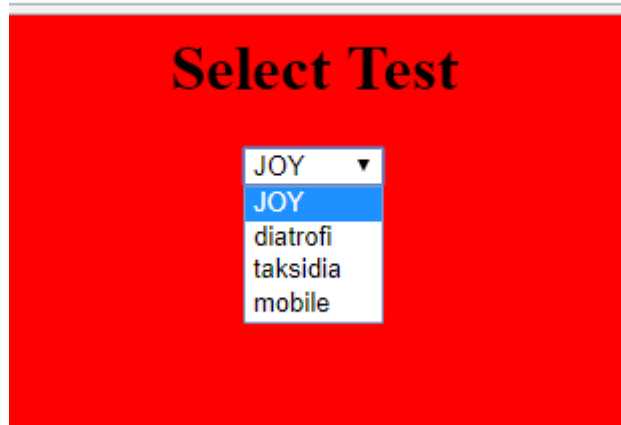
Mobile:



Εικόνα 36 Mobile's Questions and Answers

4.3 ΚΑΤΕΒΑΣΜΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ

Για να κατεβάσει κάποιος τα ερωτηματολόγια έχει 2 επιλογές. Πρώτη επιλογή: χειροκίνητα, μέσω της ιστοσελίδας <https://bill09adam01.000webhostapp.com/>. Πραγματοποιώντας σύνδεση στην ιστοσελίδα με τα στοιχεία που έχει δώσει κατά την εγγραφή (Κεφάλαιο 4.1) έχει την δυνατότητα να δει τα ερωτηματολόγια (όπως φαίνεται στην Εικόνα 37)



Εικόνα 37 Choose Questionnaire

Να διαλέξει το ερωτηματολόγιο που επιθυμεί, να το δει και να το κατεβάσει χειροκίνητα (όπως φαίνεται στην Εικόνα 38).

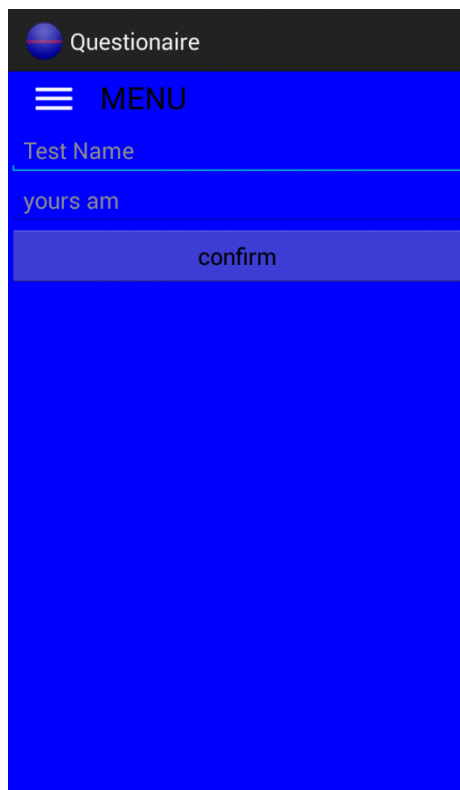


Εικόνα 38 Download Questionnaire Manually

Δεύτερη επιλογή: αυτόματα μέσω της εφαρμογής. Κατά την διαδικασία σύνδεσης, ο χρήστης επιλέγει το ερωτηματολόγιο που επιθυμεί και κάνει σύνδεση. Στην συνέχεια, εμφανίζεται το ερωτηματολόγιο προκειμένου να το απαντήσει ο χρήστης. Παρουσίαση του συγκεκριμένου τρόπου κατεβάσματος του ερωτηματολογίου θα γίνει στο επόμενο κεφάλαιο (Είσοδος στην Εφαρμογή).

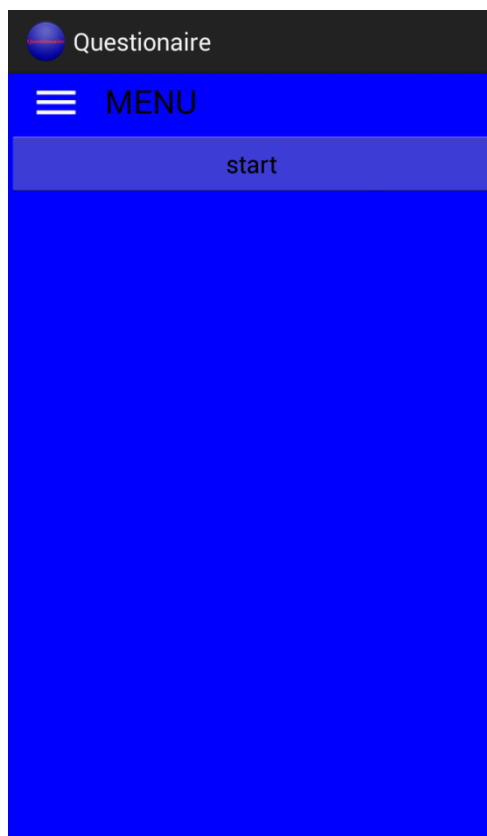
4.4 ΕΙΣΟΔΟΣ ΣΤΗΝ ΕΦΑΡΜΟΓΗ

Όταν ο χρήστης επιλέξει να τρέξει την εφαρμογή, εμφανίζεται μια φόρμα προκειμένου να επιλέξει ερωτηματολόγιο (όπως αναφέρθηκε στο κεφάλαιο κατέβασμα ερωτηματολογίων) και να κάνει σύνδεση (όπως φαίνεται στην Εικόνα 39).



Εικόνα 39 Login to Application and Download Questionnaire automatically

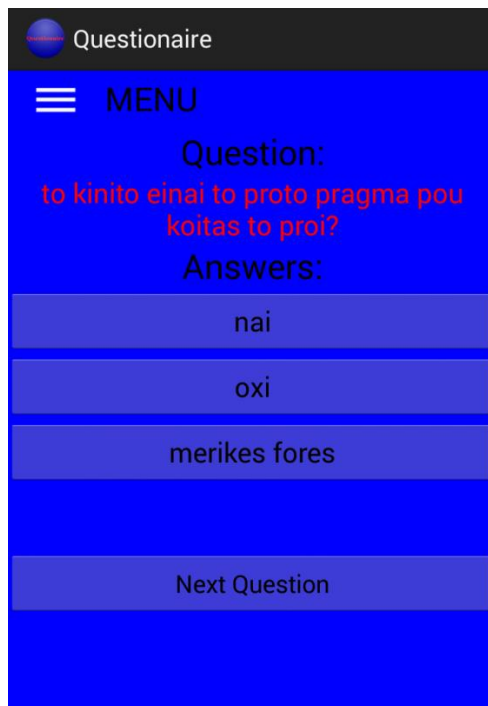
Στην συνέχεια η εφαρμογή κατεβάζει το αρχείο που απαιτείται και ενημερώνει τον χρήστη για την ορθότητα των στοιχείων. Αν η σύνδεση έχει γίνει σωστά, εμφανίζεται το κουμπί start προκειμένου να αρχίσει η διαδικασία απάντησης του ερωτηματολογίου (όπως φαίνεται στην Εικόνα 40)



Εικόνα 40 Start Questionnaire

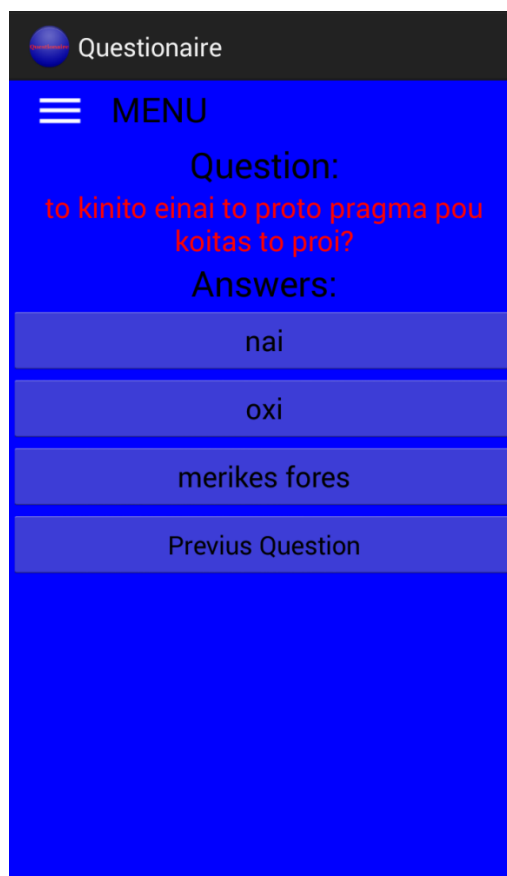
4.5 ΔΙΑΔΙΚΑΣΙΑ ΑΠΑΝΤΗΣΗΣ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ

Εφόσον ο χρήστης έχει συμπληρώσει σωστά την φόρμα σύνδεσης (ερωτηματολόγιο, όνομα χρήστη) και έχει πατήσει το πλήκτρο start, εμφανίζεται η πρώτη ερώτηση του ερωτηματολογίου και οι τρεις πιθανές απαντήσεις (οι ερωτήσεις εμφανίζονται με τυχαία σειρά) (όπως φαίνεται στην Εικόνα 41).



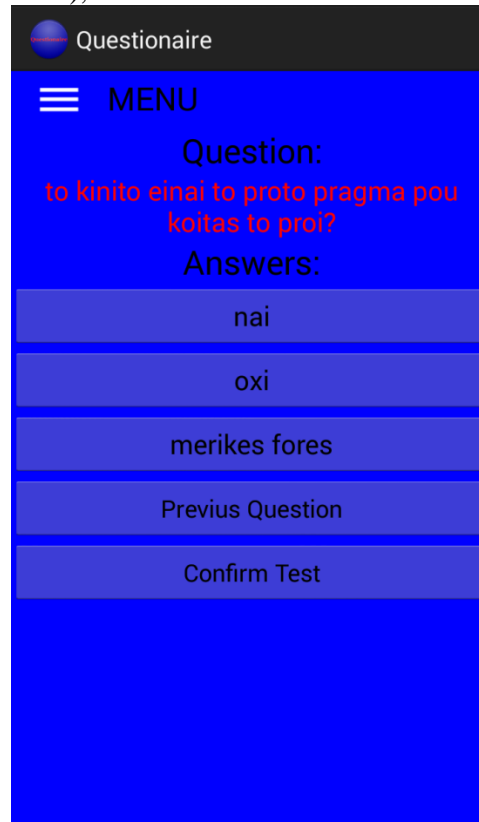
Εικόνα 41 First Question

Στην περίπτωση που υπάρχει προηγούμενη ερώτηση, εμφανίζεται το κουμπί Previous Question προκειμένου να δοθεί η δυνατότητα στον χρήστη να διορθώσει την απάντηση της προηγούμενης ερώτησης. (όπως φαίνεται στην Εικόνα 42)



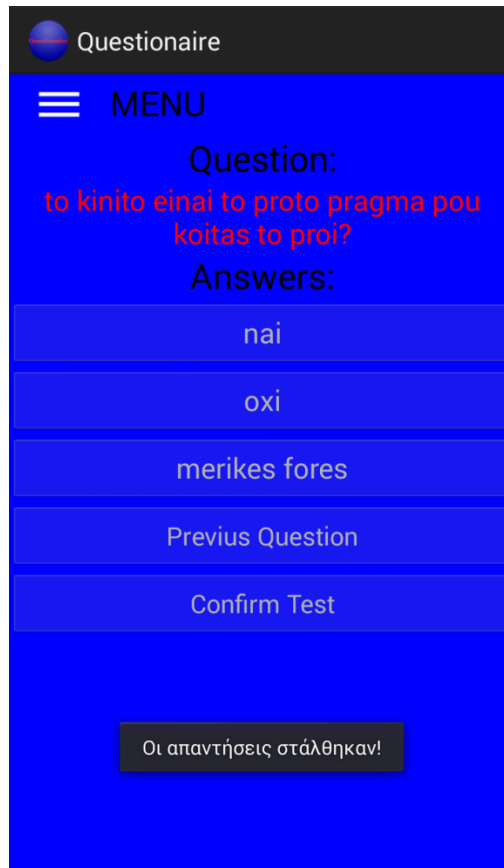
Εικόνα 42 Previous Question Available

Ενώ όταν υπάρχει επόμενη ερώτηση, εμφανίζεται και το κουμπί Next Question προκειμένου να μπορεί ο χρήστης να παραβλέψει (προς το παρόν μια απάντηση). Όταν ο χρήστης επιλέξει απάντηση, εμφανίζεται η επόμενη ερώτηση (όπως είδαμε στην εικόνα 41). Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου ο χρήστης συμπληρώσει πλήρως το ερωτηματολόγιο (το ερωτηματολόγιο θεωρείται πλήρως συμπληρωμένο όταν ο χρήστης απαντήσει όλες τις ερωτήσεις). Όταν Συμπληρωθεί πλήρως το ερωτηματολόγιο, εμφανίζεται πλήκτρο confirm (όπως φαίνεται στην Εικόνα 43),



Εικόνα 43 Confirm Question Available

με το οποίο ο χρήστης μπορεί να αποστείλει τις απαντήσεις του στην ιστοσελίδα και ενημερώνεται για την ορθή ολοκλήρωση της διαδικασίας (όπως φαίνεται στην Εικόνα 44).



Εικόνα 44 Answers Have Been Sent

4.6 ΕΞΕΛΙΞΕΙΣ ΠΟΥ ΘΑ ΜΠΟΡΟΥΣΑΝ ΝΑ ΠΡΑΓΜΑΤΟΠΟΙΗΘΟΥΝ ΣΤΟ ΜΕΛΛΟΝ

Στο κεφάλαιο αυτό θα αναφερθούμε στις εξελίξεις που θα μπορούσαν να πραγματοποιηθούν στο μέλλον με σκοπό την βελτίωση της υλοποιημένης εφαρμογής.

Κάποιες από τις βελτιώσεις που θα βοηθούσαν στο καλύτερο αποτέλεσμα είναι:

- 1) Δημιουργία κώδικα σε γλώσσα html, php, sql για την εμφάνιση αποτελεσμάτων που συμπεραίνονται από τις απαντήσεις που έχουν δώσει οι χρήστες (πχ μέσω όρο κάθε απάντησης ανά ερώτηση)
- 2) Βελτιστοποίηση κώδικα java της εφαρμογής για τον έλεγχο της σύνδεσης του χρήστη (η εφαρμογή δεν ελέγχει τα στοιχεία σύνδεσης, απλά δεν αποθηκεύει τα στοιχεία των μη εξουσιοδοτημένων χρηστών).
- 3) Καλύτερος έλεγχος για την αποφυγή δεύτερου λογαριασμού από τον ίδιο χρήστη (αυτή την στιγμή ο χρήστης δίνει έναν αλφαριθμητικό am) με την αποθήκευση επιπλέον στοιχείων κατά την δημιουργία νέου χρήστη (πχ IP address, MAC address ή αριθμό ταυτότητας).
- 4) Βελτίωση βάσης δεδομένων της εφαρμογής για την αποθήκευση ερωτήσεων και απαντήσεων, οι οποίες θα είναι γραμμένες με ελληνικούς χαρακτήρες.
- 5) Βελτίωση βάσεων δεδομένων (εφαρμογής και ιστοσελίδας) για την αποθήκευση περισσότερων από τρεις απαντήσεις ανά ερώτηση.
- 6) Βελτίωση οπτικού αποτελέσματος της εφαρμογής, προσθέτοντας περισσότερα γραφικά στοιχεία (εικόνες κλπ.).

5 ΠΑΡΑΡΤΗΜΑ (Ο ΚΩΔΙΚΑΣ)

Ο κώδικας της κλάσης MainActivity.java:

```
package com.example.questionnaire;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Random;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.os.SystemClock;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.text.Html;
import android.util.Log;
import android.view.Display;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
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.TextView;
import android.widget.Toast;
import android.widget.ViewFlipper;

@SuppressLint("UseValueOf") public class MainActivity extends Activity {
    DrawerLayout mainLayout=null;
    LinearLayout l1=null,l12=null;
    ListView list=null;
    int width=0,height=0,count=0,i=0;
    ViewFlipper flipper;
    Button answer1,answer2,answer3,back,confirm,start,next;
    EditText questionnaire,users_am;
    String am,question,test,text="";
    private Database db;
    String s[]={""};
    String q[]={""};
    int flag=0,flag1=0,exists=0;
    int end=0;
    int min = 0;
    int max = 0,i1=0;
    String ans[];
    String quest[];
    ArrayList<String> answers = new ArrayList<String>();
    ArrayList<String> questions = new ArrayList<String>();
    ArrayList<Integer> answered = new ArrayList<Integer>();
    private String
download="https://bill09adam01.000webhostapp.com/download.php?method=post&name
=";
    private String baseURL =
"https://bill09adam01.000webhostapp.com/answer.php?method=post";
    private int count1;
    private void makeHeader(String title)
    {
        LinearLayout header=new LinearLayout(this);
        l1.addView(header);
        header.setBackgroundColor(Color.BLUE);
        LinearLayout.LayoutParams
hparams=(LinearLayout.LayoutParams)header.getLayoutParams();
        hparams.width=width;
        hparams.height=height/15;
        header.setLayoutParams(hparams);
        ImageView openDrawer=new ImageView(this);
        header.addView(openDrawer);
        TextView t=new TextView(this);
        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.menuopen);
        b1=Bitmap.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.setTextSize(24);
        t.setTextColor(Color.BLACK);
        t.setText(title);
    }
    private 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.BLUE);
    }
    private void start(){
        questionnaire=new EditText(this);
        users_am=new EditText(this);
        start=new Button(this);
        LinearLayout l10=new LinearLayout(this);
        flipper.addView(l10);
        l10.setId(111);
        l10.setOrientation(LinearLayout.VERTICAL);
        l10.setBackgroundColor(Color.BLUE);
        questionnaire.setHint("Test Name");
        users_am.setHint("yours am");
        start.setText("confirm");
    }

```

```

l10.addView(questionnaire);
l10.addView(users_am);
l10.addView(start);
start.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {
        test=questionnaire.getText().toString();
        download=download+test;

        new DownloadFile().execute(download,test+".txt");
        SystemClock.sleep(1000);

        download="https://bill09adam01.000webhostapp.com/download.php?method=post
&name=";

        am=users_am.getText().toString();
        flipper.setDisplayedChild(2);
    }
});
}
private void start1(){
start=new Button(this);
start.setText("start");
answer1=new Button(this);
answer2=new Button(this);
answer3=new Button(this);
back=new Button(this);
next=new Button(this);
confirm=new Button(this);
final TextView t1=new TextView(this);
final TextView t=new TextView(this);
t1.setGravity(Gravity.CENTER);
t1.setTextSize(25);
t1.setText("Answers:");
t1.setTextColor(Color.BLACK);
final TextView question=new TextView(this);
l12=new LinearLayout(this);
flipper.addView(l12);
l12.setId(112);
l12.setOrientation(LinearLayout.VERTICAL);
l12.setBackgroundColor(Color.BLUE);
l12.addView(start);

start.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {

try {

```

```

test=test+".txt";

String x=readFromFile(test);
manageTxt(x);
exists=1;
max=count-1;
ans=new String[count];
quest=new String[count];
start.setVisibility(View.GONE);
t.setGravity(Gravity.CENTER);
t.setText("Question:");
t.setTextSize(25);
t.setTextColor(Color.BLACK);
question.setGravity(Gravity.CENTER);
question.setTextSize(20);
question.setTextColor(Color.RED);
back.setText("Previous Question");
next.setText("Next Question");
confirm.setText("Confirm Test");
s=db.getQuestions();
Random r = new Random();
i1 = r.nextInt(max - min + 1) + min;
    while(answerd.contains(i1)==true){
        i1 = r.nextInt(max - min + 1) + min;
    }
    answerd.add(i1);
q=db.getQuestionsAnswers(s[i1]);
question.setText(s[i1]);
questions.add(s[i1]);
back.setVisibility(View.INVISIBLE);
confirm.setVisibility(View.INVISIBLE);
if(i==0){i++;}
answer1.setText(q[0]);
answer2.setText(q[1]);
answer3.setText(q[2]);
answer1.setTextSize(20);
answer2.setTextSize(20);
answer3.setTextSize(20);
l12.addView(t);
l12.addView(question);
l12.addView(t1);
l12.addView(answer1);
l12.addView(answer2);
l12.addView(answer3);
l12.addView(back);
l12.addView(next);
l12.addView(confirm);

count1 = 0;
answer1.setOnClickListener(new OnClickListener()

```

```

{
    @Override
    public void onClick(View u) {
        ans[i1]=answer1.getText().toString();
        quest[i1]=question.getText().toString();
        answers.add(answer1.getText().toString());
        questions.add(question.getText().toString());
        if(answered.size()<count){
            text=text+answer1.getText()+" ";

            Random r = new Random();
            i1 = r.nextInt(max - min + 1) + min;
            while(answered.contains(i1)==true){
                i1 = r.nextInt(max - min + 1) + min;
            }
            answered.add(i1);
            question.setText(s[i1]);
            q=db.getQuestionsAnswers(s[i1]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);
            if(i>=1){back.setVisibility(View.VISIBLE);}
            if(i==count-
1){next.setVisibility(View.GONE);}
            if(i!=count-
1){next.setVisibility(View.VISIBLE);}
            }else if(i<count){
                question.setText(s[i]);
                questions.add(s[i]);
                q=db.getQuestionsAnswers(s[i]);
                answer1.setText(q[0]);
                answer2.setText(q[1]);
                answer3.setText(q[2]);
                if(i>=1){back.setVisibility(View.VISIBLE);}
                if(i==count-
1){next.setVisibility(View.GONE);}
                if(i!=count-
1){next.setVisibility(View.VISIBLE);}
            }
            i++;
            count1++;

            if(count1>=count){
                confirm.setVisibility(View.VISIBLE);
            }
        }
    }
};

```



```

answer2.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {
        ans[i1]=answer2.getText().toString();
        quest[i1]=question.getText().toString();
        answers.add(answer2.getText().toString());
        questions.add(question.getText().toString());
        if(answered.size()<count){
            text=text+answer2.getText()+" ";
            ans[i1]=answer1.getText().toString();
            quest[i1]=question.getText().toString();
            answers.add(answer2.getText().toString());
            questions.add(question.getText().toString());

            Random r = new Random();
            i1 = r.nextInt(max - min + 1) + min;
            while(answered.contains(i1)==true){
                i1 = r.nextInt(max - min + 1) + min;
            }
            answered.add(i1);
            question.setText(s[i1]);
            q=db.getQuestionsAnswers(s[i1]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);

            if(i>=1){back.setVisibility(View.VISIBLE);}
            if(i==count-
1){next.setVisibility(View.GONE);}
            if(i!=count-
1){next.setVisibility(View.VISIBLE);}

            }else if(i<count){
                question.setText(s[i]);
                questions.add(s[i]);
                q=db.getQuestionsAnswers(s[i]);
                answer1.setText(q[0]);
                answer2.setText(q[1]);
                answer3.setText(q[2]);

                if(i>=1){back.setVisibility(View.VISIBLE);}
                if(i==count-
1){next.setVisibility(View.GONE);}
                if(i!=count-
1){next.setVisibility(View.VISIBLE);}

            }
            i++;
            count1++;

            if(count1>=count){

```

```

confirm.setVisibility(View.VISIBLE);
    }
}
});
answer3.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {
        ans[i1]=answer3.getText().toString();
        quest[i1]=question.getText().toString();
        answers.add(answer3.getText().toString());
        questions.add(question.getText().toString());
        if(answered.size()<count){

            Random r = new Random();
            i1 = r.nextInt(max - min + 1) + min;
            while(answered.contains(i1)==true){
                i1 = r.nextInt(max - min + 1) + min;
            }
            answered.add(i1);
            question.setText(s[i1]);
            q=db.getQuestionsAnswers(s[i1]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);

            if(i>=1){back.setVisibility(View.VISIBLE);}
            if(i==count-
1){next.setVisibility(View.GONE);}
            if(i!=count-
1){next.setVisibility(View.VISIBLE);}

        }else if(i<count){
            question.setText(s[i]);
            questions.add(s[i]);
            q=db.getQuestionsAnswers(s[i]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);

            if(i>=1){back.setVisibility(View.VISIBLE);}
            if(i==count-
1){next.setVisibility(View.GONE);}
            if(i!=count-
1){next.setVisibility(View.VISIBLE);}

        }
        i++;
        count1++;

        if(count1>=count){

```

```

confirm.setVisibility(View.VISIBLE);
    }
}
});
back.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {
        i--;i--;
        if(i>=count-1){while(i>=count-1){i--;}}
        if(i<count){

            question.setText(s[i]);
            q=db.getQuestionsAnswers(s[i]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);
            back.setVisibility(View.INVISIBLE);

            if(i>=1){back.setVisibility(View.VISIBLE);}
            if(i==count-
1){next.setVisibility(View.GONE);}
            if(i!=count-
1){next.setVisibility(View.VISIBLE);}
        }

        i++;
        count1++;

        count1--;

    }
});
next.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {

        if(i<count){

            question.setText(s[i]);
            q=db.getQuestionsAnswers(s[i]);
            answer1.setText(q[0]);
            answer2.setText(q[1]);
            answer3.setText(q[2]);
            back.setVisibility(View.INVISIBLE);
            if(i==count-
1){back.setVisibility(View.VISIBLE);}

```

```

1){next.setVisibility(View.GONE);}
1){next.setVisibility(View.VISIBLE);}
}

        i++;

        count1++;

    }
});
confirm.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View u) {
        text=text+answer3.getText()+".";
        ans[i1]=answer1.getText().toString();
        quest[i1]=question.getText().toString();
        answers.add(answer3.getText().toString());
        answer1.setEnabled(false);
        answer2.setEnabled(false);
        answer3.setEnabled(false);
        back.setEnabled(false);
        confirm.setEnabled(false);
        next.setEnabled(false);

        end();

    }
});

} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();

    flipper.setDisplayedChild(0);

}
});
}

private void credits()
{
    TextView info=new TextView(this);
    flipper.addView(info);
    info.setText(Html.fromHtml("<h3>Credits</h3>Created
by:<br><br>Tzimas Vasileios A.M 13978<br><br>I Hope You Will Enjoy My
Application!!"));
    info.setGravity(Gravity.CENTER);

```

```

        info.setTextSize(20);
        info.setTextColor(Color.RED);
    }
    private void end()
    {

        for(int i=0;i<count;i++)
        {

            try
            {

                String url = baseUrl + "&question=" +
                URLEncoder.encode(quest[i],"UTF-8") + "&user=" + URLEncoder.encode(am,"UTF-
                8")+ "&test="+URLEncoder.encode(test,"UTF-
                8")+ "&answer="+URLEncoder.encode(ans[i],"UTF-8");

                HttpAsyncTask hat = new HttpAsyncTask();
                hat.execute(url);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
        if(haveNetworkConnection()==true){
            Toast.makeText(MainActivity.this,"Οι απαντήσεις
            στάλθηκαν!",Toast.LENGTH_SHORT).show();
        }

    }
    private void makeList()
    {
        ArrayList<String> s=new ArrayList<String>();
        s.add("Questionnaire");
        s.add("Credits");
        ArrayAdapter<String> adapter=new
        ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,s);
        list=new ListView(this);
        TextView padding=new TextView(this);
        padding.setHeight(height/20);
        list.addHeaderView(padding);
        padding.setBackgroundColor(Color.BLUE);
        padding.setTextColor(Color.BLACK);
        padding.setTextSize(20);
        padding.setText("Make a choice");
        padding.setGravity(Gravity.CENTER);
    }
}

```

```

        list.setAdapter(adapter);
        mainLayout.addView(list);
        DrawerLayout.LayoutParams
cparams=(DrawerLayout.LayoutParams)list.getLayoutParams();
        cparams.width=5*width/5;
        cparams.height=DrawerLayout.LayoutParams.MATCH_PARENT;
        cparams.gravity=Gravity.START;
        list.setLayoutParams(cparams);
        list.setAdapter(adapter);
        list.setBackgroundColor(Color.BLUE);
        list.setOnItemClickListener(new OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
                int position, long id) {
                if(position==0) return ;
                flipper.setDisplayedChild(position-1);
                mainLayout.closeDrawer(list);
            }
        });
    }
    @SuppressWarnings("deprecation")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mainLayout=new DrawerLayout(this);
        mainLayout.setDrawerShadow(R.drawable.drawershadow, GravityCompat.START);
        this setContentView(mainLayout);
        Display display = getWindowManager().getDefaultDisplay();
        width = display.getWidth();
        height = display.getHeight();
        flipper=new ViewFlipper(this);
        db=new Database(this, "users.db", null, 1);
        db.clearData();
        count=0;
        text="";
        answered.clear();
        i1=0;
        makeHome();
        makeList();
        makeHeader("MENU");
        l1.addView(flipper);

        start();
        credits();
        start1();
    }
    private String readFromFile(String x) throws IOException {

```

```

File path =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
File file = new File(path, x);
int length = (int) file.length();

byte[] bytes = new byte[length];

FileInputStream in = new FileInputStream(file);
try {
    in.read(bytes);
} finally {
    in.close();
}
String contents = new String(bytes);
return contents;
}
private void manageTxt(String x) throws IOException{
    char a[]=x.toCharArray();
    int flag=0;

    question="";
    test="";
    int sum=0;
    for(int i=0;i<x.length();i++)
        if(a[i]=='.')
            sum++;
    sum=sum-2;
    String answer="",answer1="",answer2="";
    for(int i=0;i<x.length();i++){
        if(a[i]=='.' && flag==0)
            flag++;
        else if(flag==1 && a[i]!='.'){
            am=a[i];
            if(a[i]=='.' && flag==1)
                flag++;
            if(a[i]=='.' && flag==2)
                flag++;
            else if(flag==3 && a[i]!='.'){
                question=question+a[i];
                if(a[i]=='.' && flag==3)
                    flag++;
                if(a[i]=='.' && flag==4)
                    flag++;
                else if(flag==5 && a[i]!='.'){
                    answer=answer+a[i];
                    if(a[i]=='.' && flag==5)
                        flag++;
                    if(a[i]=='.' && flag==6)
                        flag++;

```

```

else if(flag==7&&a[i]!='.')
    answer1=answer1+a[i];
if(a[i]=='.'&&flag==7)
    flag++;
if(a[i]=='.'&&flag==8)
    flag++;
else if(flag==9&&a[i]!='.')
    answer2=answer2+a[i];
if(a[i]=='.'&&flag==9){
    sum--;
    flag++;
}
if(flag==10&&sum>0){
    db.insert(question, answer, answer1, answer2);
    count++;
    flag=2;
    question="";
    answer="";
    answer1="";
    answer2="";
}
else{
    if(a[i]=='.'&&flag==10)
        flag++;
    else if(flag==11&&a[i]!='.')
        test=test+a[i];
    if(a[i]=='.'&&flag==11)
        flag++;
}
}
if(question==""||a[0]=='<'){
    throw new IOException("problem");
}
db.insert(question, answer, answer1, answer2);
count++;
text=am+", ";
text=text+test+", ";
}

```

```
private class HttpAsyncTask extends AsyncTask<String, Void, String> {
```

```

@Override
protected String doInBackground(String... urls) {

    return httpRequestResponse(urls[0]);
}
// onPostExecute displays the results of the AsyncTask.
@Override
protected void onPostExecute(String result) {

```



```
}  
}
```

//For HttpAsync Functions: sending requests and receiving responses

```
public static String httpRequestResponse(String url){  
    InputStream inputStream = null;  
    String result = "";  
    try {  
        // create HttpClient  
        HttpClient httpClient = new DefaultHttpClient();  
  
        // make GET request to the given URL  
        HttpResponse httpResponse = httpClient.execute(new HttpGet(url));  
  
        // receive response as inputStream  
        inputStream = httpResponse.getEntity().getContent();  
  
        // convert InputStream to string  
        if(inputStream != null)  
            result = convertInputStreamToString(inputStream);  
        else  
            result = "InputStream did not work";  
  
    } catch (Exception e) {  
        Log.d("InputStream", e.getMessage());  
    }  
  
    return result;  
}
```

private static String convertInputStreamToString(InputStream inputStream) throws

```
IOException{  
    BufferedReader bufferedReader = new BufferedReader( new  
    InputStreamReader(inputStream));  
    String line = "";  
    String result = "";  
    while((line = bufferedReader.readLine()) != null)  
        result += line;  
  
    inputStream.close();  
    return result;  
}
```

private class DownloadFile extends AsyncTask<String, Void, Void>{

```
@Override  
protected Void doInBackground(String... strings) {  
    String fileUrl = strings[0];  
    String fileName = strings[1];
```

```

        File path =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLO
ADS);
        path.mkdir();
        File pdfFile = new File(path, fileName);

        try{
            pdfFile.createNewFile();
        }catch (IOException e){
            e.printStackTrace();
        }
        FileDownloader.downloadFile(fileUrl, pdfFile);

        return null;
    }
}
private boolean haveNetworkConnection() {
    boolean haveConnectedWifi = false;
    boolean haveConnectedMobile = false;

    ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo[] netInfo = cm.getAllNetworkInfo();
    for (NetworkInfo ni : netInfo) {
        if (ni.getTypeName().equalsIgnoreCase("WIFI"))
            if (ni.isConnected())
                haveConnectedWifi = true;
        if (ni.getTypeName().equalsIgnoreCase("MOBILE"))
            if (ni.isConnected())
                haveConnectedMobile = true;
    }
    return haveConnectedWifi || haveConnectedMobile;
}
}
}

```

Ο κώδικας της κλάσης Database.java:

```
package com.example.questionnaire;

import java.util.ArrayList;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteOpenHelper;

public class Database extends SQLiteOpenHelper{
    public int count=0;
    private SQLiteDatabase database;
    public Database(Context context, String dname, CursorFactory factory,
        int version) {
        super(context, dname, factory, version);
        database=this.getWritableDatabase();
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table questions(question text,answer text,answer1
text,answer2 text)");
    }
    public ArrayList<Question> getResults()
    {
        ArrayList<Question> x=new ArrayList<Question>();
        Cursor cursor=database.rawQuery("select * from questions",null);
        if(cursor.getCount()>0)
        {
            cursor.close();
            return x;
        }
        int questionindex=cursor.getColumnIndex("question");
        int answerindex=cursor.getColumnIndex("answer");
        int answer1index=cursor.getColumnIndex("answer1");
        int answer2index=cursor.getColumnIndex("answer2");
        cursor.moveToFirst();
        do
        {
            Question s=new Question(cursor.getString(questionindex),
                cursor.getString(answerindex),
                cursor.getString(answer1index),
                cursor.getString(answer2index));
            x.add(s);
        }while(cursor.moveToNext());
        cursor.close();
        return x;
    }
}
```

```

public String[] getQuestions()
{
    Cursor cursor=database.rawQuery("select question from questions",null);
    String s[]= new String[cursor.getCount()];
    count=cursor.getCount();
    if(cursor.getCount()==0)
    {
        cursor.close();
        s[0]="";
        return s;
    }
    int questionindex=cursor.getColumnIndex("question");
    int i=0;
    cursor.moveToFirst();
    do
    {
        String s1=cursor.getString(questionindex);
        s[i]=s1;
        i++;
    }while(cursor.moveToNext());
    cursor.close();
    return s;
}
public String[] getQuestionsAnswers(String question){
    Cursor cursor=database.rawQuery("select answer,answer1,answer2 from
questions where question='"+question+"'",null);
    String s[]= new String[3];
    if(cursor.getCount()==0)
    {
        cursor.close();
        s[0]="";
        return s;
    }
    int answerindex=cursor.getColumnIndex("answer");
    int answer1index=cursor.getColumnIndex("answer1");
    int answer2index=cursor.getColumnIndex("answer2");
    cursor.moveToFirst();
    do
    {
        s[0]=cursor.getString(answerindex);
        s[1]=cursor.getString(answer1index);
        s[2]=cursor.getString(answer2index);
    }while(cursor.moveToNext());
    cursor.close();
    return s;
}
public void insert(String question,String answer,String answer1,String answer2)
{
    database.execSQL("insert into questions(question,answer,answer1,answer2)
values("+

```

```
        question+"", "+answer+", "+answer1+", "+answer2+"));
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop table questions");
        onCreate(db);
    }
    public void clearData()
    {
        database.execSQL("delete from questions");
    }
}
```

Ο κώδικας της κλάσης FileDownloader.java:

```
package com.example.questionnaire;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class FileDownloader {
    private static final int MEGABYTE = 1024 * 1024;

    public static void downloadFile(String fileUrl, File directory){
        try {

            URL url = new URL(fileUrl);
            HttpURLConnection urlConnection = (HttpURLConnection)url.openConnection();
            //urlConnection.setRequestMethod("GET");
            //urlConnection.setDoOutput(true);
            urlConnection.connect();

            InputStream inputStream = urlConnection.getInputStream();
            FileOutputStream fileOutputStream = new FileOutputStream(directory);
            int totalSize = urlConnection.getContentLength();

            byte[] buffer = new byte[MEGABYTE];
            int bufferLength = 0;
            while((bufferLength = inputStream.read(buffer))>0 ){
                fileOutputStream.write(buffer, 0, bufferLength);
            }
            fileOutputStream.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

[57]

Ο Κώδικας της κλάσης Question.java

```
package com.example.questionnaire;
```

```
public class Question {  
    public String question;  
    public String answer;  
    public String answer1;  
    public String answer2;  
    public Question(String q,String n,String p,String c)  
    {  
        question=q;  
        answer=n;  
        answer1=p;  
        answer2=c;  
    }  
}
```

6 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]

https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Assembly_languages

[2]

https://en.wikipedia.orghttp://bitsavers.trailing-edge.com/pdf/ibm/140x/C24-3319-0_1401_tapeAutocod.pdf/wiki/List_of_programming_languages_by_type#Assembly_languages

[3]

<http://www.linuxjournal.com/article/8408>

[4]

<http://archive.computerhistory.org/resources/text/Fortran/102663110.05.01.acc.pdf>

[5]

[http://en.wikipedia.org/wiki/X_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/X_Sharp_(programming_language))

[6]

<http://en.wikipedia.org/wiki/MACRO-11>

[7]

http://en.wikipedia.org/wiki/GNU_Assembler#Usage

[8]

<http://www.chem.ox.ac.uk/fortran/fortran1.html>

[9]

<http://www.webopedia.com/TERM/C/COBOL.html>

[10]

[http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Lisp_\(programming_language\).html](http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Lisp_(programming_language).html)

[11]

http://www.tutorialspoint.com/cprogramming/c_overview.htm

[12]

[http://en.wikipedia.org/wiki/Pascal_\(programming_language\)](http://en.wikipedia.org/wiki/Pascal_(programming_language))

[13]

<http://en.wikipedia.org/wiki/C%2B%2B>

[14]

[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

[15]

<http://www.it.uom.gr/project/java/tutorial.htm#cla>

[16]

<http://delphi.about.com/od/beginners/a/dbeginner6.htm>

[17]

<http://www.atozed.com/CrossTalk/Blog/20090505.en.aspx>

[18]

<https://www.programiz.com/python-programming>

[19]

<https://www.thoughtco.com/what-is-visual-basic-3423998>

[20]

<http://searchwindevelopment.techtarget.com/definition/C>

[21]

https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Assembly_languages

[22]

https://el.wikipedia.org/wiki/%CE%9A%CE%B9%CE%BD%CE%B7%CF%84%CF%8C_%CF%84%CE%B7%CE%B%CE%AD%CF%86%CF%89%CE%BD%CE%BF

[23]

<https://www.sansimera.gr/articles/241>

[24]

<https://www.xcubelabs.com/mobile-operating-systems/>

[25]

<http://www.greeceandroid.gr/dev/811-smartphone-wars>

[26]

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#Alpha

[27]

https://el.wikipedia.org/wiki/Android_Donut

[28]

https://el.wikipedia.org/wiki/Android_Eclair

[29]

https://el.wikipedia.org/wiki/Android_Froyo

[30]

https://el.wikipedia.org/wiki/Android_Gingerbread

[31]

https://el.wikipedia.org/wiki/Android_Honeycomb

[32]

https://el.wikipedia.org/wiki/Android_Ice_Cream_Sandwich

[33]

https://el.wikipedia.org/wiki/Android_Jelly_Bean

[34]

https://el.wikipedia.org/wiki/Android_KitKat

[35]

https://el.wikipedia.org/wiki/Android_Lollipop

[36]

https://el.wikipedia.org/wiki/Android_Marshmallow

[37]

https://el.wikipedia.org/wiki/Android_Nougat

[38]

<http://www.pocket-lint.com/news/141195-when-is-android-oreo-coming-to-my-phone>

[39]

<https://el.wikipedia.org/wiki/Apple>

- [40]
https://en.wikipedia.org/wiki/IOS_version_history#iOS_11
- [41]
https://en.wikipedia.org/wiki/Windows_Mobile
- [42]
<https://www.moontechnolabs.com/apple-vs-android-comparative-study-2017/>
- [43]
<http://coolweb.gr/leitourgika-systimata-kinton-smartphone/>
- [44]
<http://blog.appleworldhellas.com/%CE%AC%CF%80%CE%BF%CF%88%CE%B7-ios-vs-android-%CE%BA%CE%BB%CE%B5%CE%B9%CF%83%CF%84%CF%8C-vs-%CE%B1%CE%BD%CE%BF%CE%B9%CF%87%CF%84%CF%8C-%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CE%BC%CE%B9%CE%BA%CF%8C/>
- [45]
<http://www.greeceandroid.gr/dev/568-how-to-upload-to-google-play>
- [46]
<https://clearbridgemobile.com/how-to-submit-an-app-to-the-app-store/>
- [47]
<http://www.c-sharpcorner.com/UploadFile/1ae37f/how-to-upload-your-app-to-windows-store-step-by-step/>
- [48]
<http://www.instructables.com/id/How-To-Setup-Eclipse-for-Android-App-Development/>
- [49]
<http://theopentutorials.com/tutorials/android/how-to-create-android-avd-emulator-in-eclipse/>
- [50]
<https://www.ibm.com/developerworks/library/x-andbene1/index.html>
- [51]
<https://el.wikipedia.org/wiki/HTML>
- [52]
<http://groups.engin.umd.umich.edu/CIS/course.des/cis400/html/hellow.html>
- [53]
<https://el.wikipedia.org/wiki/PHP>
- [54]
<http://php.net/manual/en/tutorial.firstpage.php>
- [55]
<https://el.wikipedia.org/wiki/JavaScript>
- [56]
https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [57]
<https://stackoverflow.com/questions/24740228/android-download-pdf-from-url-then-open-it-with-a-pdf-reader>