



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και ανάπτυξη mobile εφαρμογής του
ΤΗΜΜΥ του Πανεπιστημίου Θεσσαλίας
σε περιβάλλον Android**

**Design and implementation of DECE/UTH
Android application**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Γεωργιάδη

Βόλος, Ιούλιος 2015

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

**Σχεδιασμός και ανάπτυξη mobile εφαρμογής του
ΤΗΜΜΥ του Πανεπιστημίου Θεσσαλίας
σε περιβάλλον Android**

**Design and implementation of DECE/UTH
Android application**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Γεωργιάδη

Επιβλέποντες :

Παναγιώτα Τσομπανοπούλου
Επίκουρος Καθηγήτρια Π.Θ.

Παναγιώτης Μποζάνης
Καθηγητής Π.Θ.

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την ημερομηνία εξέτασης.

(Υπογραφή)

.....
Παναγιώτα Τσομπανοπούλου
Επίκουρος Καθηγήτρια Π.Θ.

(Υπογραφή)

.....
Παναγιώτης Μποζάνης
Καθηγητής Π.Θ.

Βόλος, Ιούλιος 2015

(Υπογραφή)

.....

Χρήστος Γεωργιάδης

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων
του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστημίου
Θεσσαλίας

© 2015– All rights reserved

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Με την περάτωση της παρούσας διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντές μου, κα Παναγιώτα Τσομπανοπούλου και κο Παναγιώτη Μποζάνη, για την εμπιστοσύνη που επέδειξαν στο πρόσωπό μου, καθώς και την πολύτιμη καθοδήγησή τους καθ' όλη τη διάρκεια αυτής της περιόδου.

Επίσης, θα ήθελα να ευχαριστήσω τον κο Αθανάσιο Φεύγα για τις εύστοχες παρατηρήσεις του κατά τη διάρκεια υλοποίησης της εφαρμογής, καθώς και τους κους Νικόλαο Φραγγογιάννη και Απόστολο Τσιόβουλο για την βοήθειά τους σε διάφορα τεχνικά προβλήματα που προέκυψαν.

Ένα μεγάλο ευχαριστώ οφείλω να πω στους φίλους μου, Τρύφωνα και Ανδρέα, για την επιμονή και τις πολύτιμες συμβουλές τους κατά τη διάρκεια αυτής της προσπάθειας. Ακόμα ένα μεγάλο ευχαριστώ στον αδερφό μου, Δημήτρη, ο οποίος έχει μεγάλο μερίδιο ευθύνης στο σχεδιασμό του γραφικού περιβάλλοντος της εφαρμογής.

Ακόμα, θέλω να ευχαριστήσω τους φίλους μου Ειρήνη, Ιωάννα, Πέλια, Μιχάλη, Νίκο και Ομέρ για την δοκιμή της εφαρμογής στις προσωπικές τους συσκευές.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, την Άννα και όλους τους ανθρώπους που όλα αυτά τα χρόνια των σπουδών μου ήταν δίπλα μου και με στήριζαν με κάθε τρόπο.

Περίληψη

Η ραγδαία αύξηση και η καθημερινή χρήση έξυπνων τηλεφώνων (smartphones) στις μέρες μας, οδηγεί στην ανάγκη ανάπτυξης όλο και περισσότερων εφαρμογών που έχουν ως κύριο στόχο την παροχή εξειδικευμένων υπηρεσιών στο χρήστη.

Η παρούσα εργασία έχει ως στόχο την ανάπτυξη μιας εφαρμογής σε λειτουργικό σύστημα Android, η οποία θα χρησιμοποιείται από τους προπτυχιακούς φοιτητές του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας. Η εφαρμογή θα προσφέρει μια πληθώρα επιλογών στο χρήστη, όπως προβολή των ανακοινώσεων του τμήματος, προβολή προσωπικών βαθμών, προβολή και διαμόρφωση του εβδομαδιαίου προγράμματος μαθημάτων, κ.α.

Για τις ανάγκες της εφαρμογής πέρα από την υλοποίησή της, δημιουργήθηκε ένας Apache εξυπηρέτης (server), ο οποίος χρησιμοποιεί κρυπτογραφημένη επικοινωνία με την εφαρμογή μέσω του πρωτοκόλλου https, καθώς επίσης και μία βάση δεδομένων, η οποία φιλοξενείται στον εξυπηρέτη.

Τέλος, η εφαρμογή έχει δοκιμαστεί σε πραγματικές συσκευές, με τα αποτελέσματα να είναι άκρως ικανοποιητικά.

Abstract

The rapid growth and the daily use of smartphones nowadays, leads to the need for more and more application development that primarily focus on providing specialized services to the user.

The present thesis aims to develop an Android application which will be used by the undergraduate students of DECE/UTH. The application will offer a wealth of options to the user, such as department's announcements, personal grades, presentation and configuration of the weekly curriculum, etc.

For the application's needs beyond the implementation, an https Apache server which uses entirely encrypted communication with the application and a database built in the server were created.

Finally, the application has been tested on real devices with highly satisfactory results.

Περιεχόμενα

1	Εισαγωγή	15
1.1	Σκοπός εργασίας	15
1.2	Κίνητρο	15
	Παγκόσμιοι χρήστες έξυπνων τηλεφώνων	15
	Mobile vs PC	16
	Πανεπιστημιακές mobile εφαρμογές	18
1.3	Δομή εργασίας	18
2	Υπόβαθρο	19
2.1	Το λειτουργικό σύστημα Android	19
	Η εξέλιξη του Android	19
	Εφαρμογές Android	21
	Υλικό	21
	Αρχιτεκτονική	21
	Ασφάλεια	24
2.2	XAMPP	25
	Ετυμολογία	25
	Δυνατότητες και απαιτήσεις	25
	Χρήση	25
	Χαρακτηριστικά	26
2.3	Apache HTTP server	26
2.4	PHP	27
	Ιστορία της PHP	27
2.5	MySQL	28
2.6	Το πρωτόκολλο επικοινωνίας HTTPS	28
	Τρόπος λειτουργίας	28
2.7	Wireshark	29
3	Αρχιτεκτονική συστήματος	30
3.1	Σχέδιο υλοποίησης	30
3.2	Ανάλυση υπομονάδων	31
	Android εφαρμογή	31
	Εξυπνήτης (Server)	31

3.3	Λογική σχεδιασμού.....	32
	Γιατί Android;	32
	Γιατί εξυπηρέτης;	32
	Ασφάλεια επικοινωνίας με τον εξυπηρέτη	33
4	Ανάλυση υπομονάδων συστήματος.....	34
4.1	Android εφαρμογή.....	34
	Γενική περιγραφή	35
	Περιγραφή των activities	38
	Λογική σχεδιασμού	54
4.2	Εξυπηρέτης.....	55
	Γενική περιγραφή	56
	Περιγραφή των ετικετών (tags)	59
5	Διεπαφή χρήστη (UI) εφαρμογής.....	66
5.1	Εικονίδιο εφαρμογής	66
5.2	Εικόνα εισόδου & αρχική σελίδα της εφαρμογής.....	66
5.3	Ανακοινώσεις.....	67
5.4	Διδάσκοντες.....	68
5.5	Βαθμοί	69
5.6	Χάρτης.....	69
5.7	Πρόγραμμα	70
5.8	Μαθήματα.....	72
5.9	Σίτιση.....	73
5.10	Λεωφορεία.....	74
5.11	Φυσιογνωμία.....	74
5.12	Σύνδεση	74
6	Ανάπτυξη εφαρμογής, απαιτήσεις συστήματος & δοκιμές	76
6.1	Ανάπτυξη εφαρμογής.....	76
	Android εφαρμογή	76
	Εξυπηρέτης	76
6.2	Απαιτήσεις συστήματος.....	76
6.3	Δοκιμές (Beta testing).....	77
7	Επίλογος.....	78

7.1	Συμπεράσματα	78
7.2	Μελλοντική επέκταση	78
8	Βιβλιογραφία	79

1 Εισαγωγή

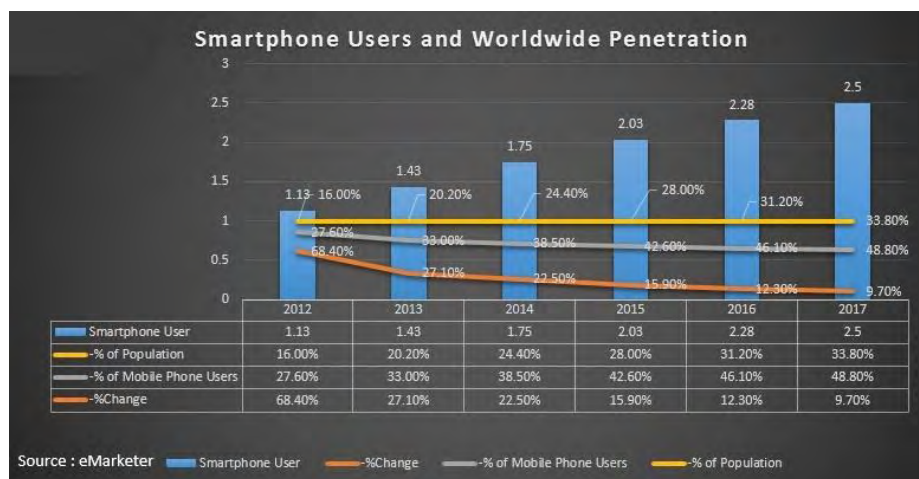
1.1 Σκοπός εργασίας

Η εργασία αυτή εστιάζει στην ανάπτυξη mobile εφαρμογής σε λειτουργικό σύστημα Android για τους προπτυχιακούς φοιτητές του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας. Συγκεκριμένα, η εφαρμογή έχει ως κύριο στόχο να απλοποιήσει την ενδοπανεπιστημιακή καθημερινότητα των φοιτητών.

1.2 Κίνητρο

Δύο ήταν τα βασικά κίνητρα για την ανάπτυξη της εν λόγω εφαρμογής. Πρώτον, η ραγδαία εξάπλωση των έξυπνων τηλεφώνων τα τελευταία χρόνια, καθώς και ο συνεχώς αυξανόμενος αριθμός χρηστών Internet από αυτά. Δεύτερον, από μια μικρή έρευνα, η οποία πραγματοποιήθηκε, διαπιστώθηκε η έλλειψη εφαρμογών για τα ελληνικά πανεπιστήμια σε αντίθεση με ότι συμβαίνει στο εξωτερικό. Οι παρακάτω εικόνες επιβεβαιώνουν αυτούς τους ισχυρισμούς ^{[1][2][3][4]}.

Παγκόσμιοι χρήστες έξυπνων τηλεφώνων



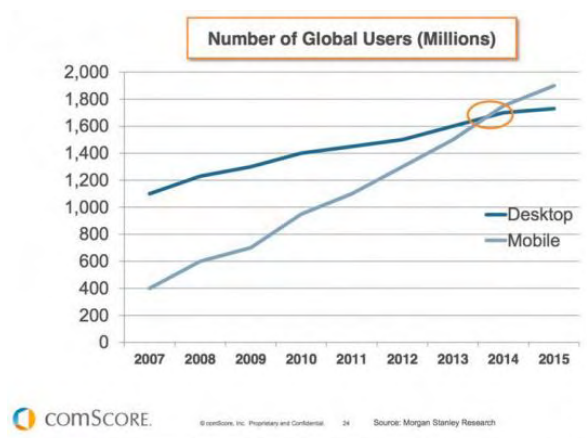
Εικόνα 1.2.1: Παγκόσμιοι χρήστες έξυπνων τηλεφώνων

Όπως φαίνεται και στην εικόνα 1.2.1, στα τέλη του 2014 ο αριθμός των παγκόσμιων χρηστών έξυπνων τηλεφώνων έφτασε τα 1.75 δισεκατομμύρια, δηλαδή περίπου στο 25% του παγκόσμιου πληθυσμού. Ο αριθμός αυτός αναμένεται να αυξηθεί

στα 2.5 δισεκατομμύρια στα τέλη του 2017 αγγίζοντας το 34% του παγκόσμιου πληθυσμού.

Mobile vs PC

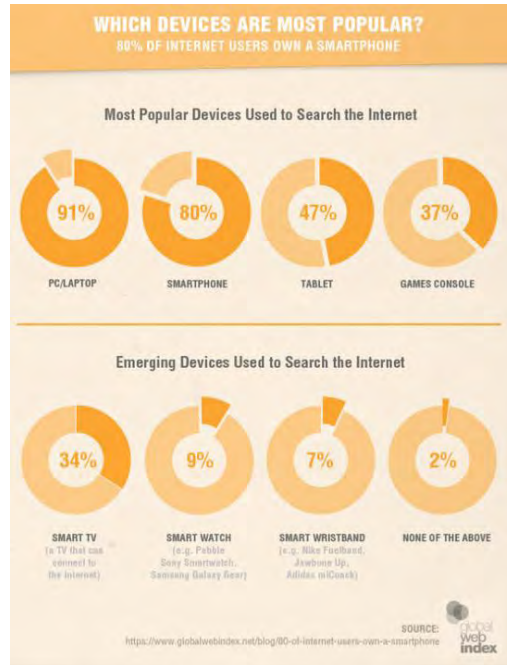
Η εικόνα που ακολουθεί, παρουσιάζει την τάση που επικρατεί τα τελευταία χρόνια. Όπως εύκολα μπορεί να παρατηρήσει κάποιος, το 2014 η πρόσβαση στο Internet μέσω mobile συσκευών ξεπέρασε τη πρόσβαση στο Internet μέσω Η/Υ. Και οι προβλέψεις για το μέλλον αναφέρουν ότι η διαφορά αυτή θα αυξηθεί περισσότερο.



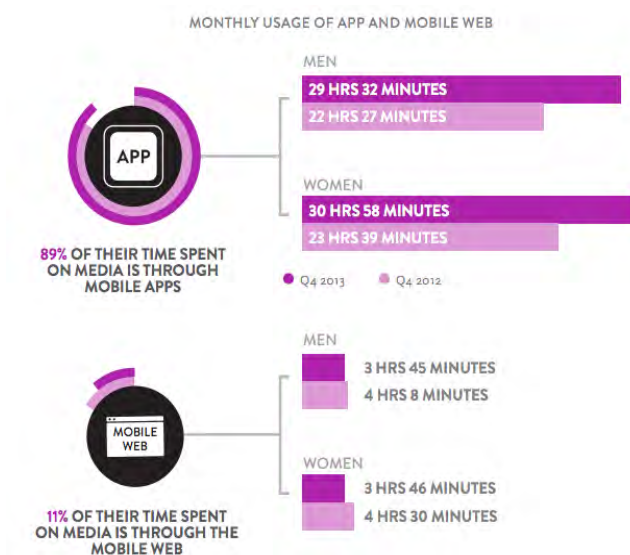
Εικόνα 1.2.2: Παγκόσμιοι χρήστες Internet

Στην εικόνα 1.2.3 φαίνεται ότι το 80% των χρηστών Internet κατέχει κάποιο έξυπνο τηλέφωνο, ενώ το 91% των χρηστών κατέχει υπολογιστή. Ο συνδυασμός αυτών των δύο εικόνων οδηγεί στο συμπέρασμα ότι οι χρήστες προτιμούν τη πλοήγηση στο Internet μέσω των έξυπνων συσκευών τους παρά μέσω του υπολογιστή.

Τέλος, στην εικόνα 1.2.4 είναι ξεκάθαρο ότι οι χρήστες προτιμούν σχεδόν απόλυτα τις mobile εφαρμογές σε σχέση με το mobile web. Σε μία μηνιαία μέτρηση που πραγματοποιήθηκε η χρήση mobile εφαρμογών έφτασε σε ποσοστό 89%. Κάτι τέτοιο λοιπόν καθιστά σαφές ότι η ανάπτυξη mobile εφαρμογών είναι η νέα τάση στην αγορά των έξυπνων τηλεφώνων και ταμπλετών.



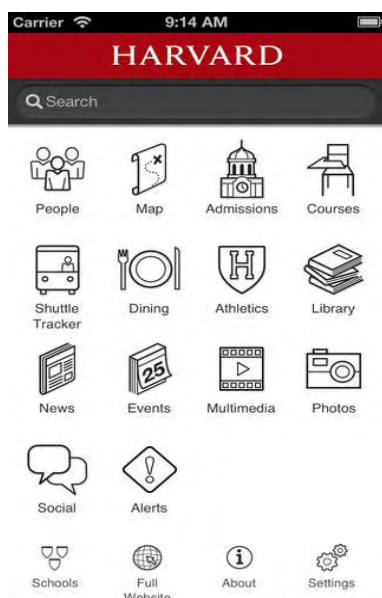
Εικόνα 1.2.3: Συσκευές που προτιμούν οι χρήστες του Internet



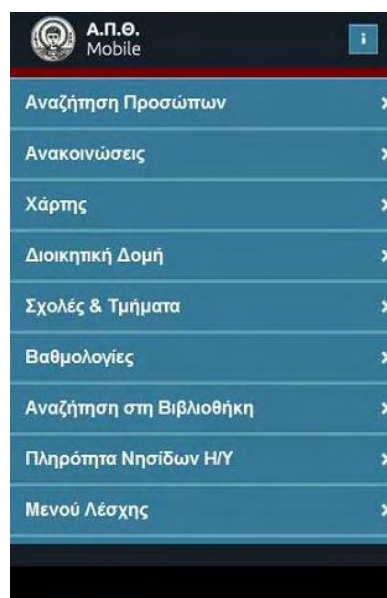
Εικόνα 1.2.4: Μηνιαία χρήση mobile app vs mobile web

Πανεπιστημιακές mobile εφαρμογές

Όπως αναφέρθηκε και στην παράγραφο 1.2, στην Ελλάδα δεν υπάρχουν πολλές πανεπιστημιακές mobile εφαρμογές. Συγκεκριμένα, από μια μικρή έρευνα που έγινε στο Play Store της Google, μόνο το Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης διαθέτει μέχρι και σήμερα επίσημη εφαρμογή για έξυπνα τηλέφωνα (εικόνα 1.2.6). Αντιθέτως, αν κάποιος αναζητήσει mobile εφαρμογές ξένων ιδρυμάτων, θα διαπιστώσει ότι αρκετά από τα πιο γνωστά πανεπιστήμια του εξωτερικού διαθέτουν κάποια. Ενδεικτικά αναφέρουμε το University of Phoenix, Newcastle University, Texas A&M University, MIT, Harvard University, κ.α (εικόνα 1.2.5). Η ανάπτυξη λοιπόν μίας τέτοιας mobile εφαρμογής κρίνεται αναγκαία και για τα ελληνικά πανεπιστήμια.



Εικόνα 1.2.5: Harvard mobile



Εικόνα 1.2.6: ΑΠΘ mobile

1.3 Δομή εργασίας

Στο 2^ο κεφάλαιο που ακολουθεί παρουσιάζεται το θεωρητικό υπόβαθρο για την ανάπτυξη της εν λόγω εφαρμογής. Το 3^ο κεφάλαιο αναλύει την αρχιτεκτονική του συστήματος σε υψηλό επίπεδο (high-level diagram). Το 4^ο κεφάλαιο αποτελεί την περιγραφή όλων των λεπτομερειών πίσω από την υλοποίηση. Στο 5^ο κεφάλαιο παρουσιάζεται η εφαρμογή από την οπτική πλευρά του χρήστη. Στο 6^ο κεφάλαιο αναλύεται το σύστημα, στο οποίο αναπτύχθηκε η εφαρμογή και ο εξυπηρέτης. Επίσης, παρουσιάζονται οι απαιτήσεις για την εγκατάσταση και τη χρήση της εφαρμογής, αλλά και διάφορες δοκιμές που έλαβαν χώρα. Τέλος, τα κεφάλαια 7 και 8 φιλοξενούν τον επίλογο και τη βιβλιογραφία αντίστοιχα.

2 Υπόβαθρο

2.1 Το λειτουργικό σύστημα Android

Το **Android**^[5] είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο χρησιμοποιεί τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την [Handset Alliance | Open Handset Alliance]. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και οι ταμπλέτες, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί και σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής υλικού (hardware), οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Irina Blok.



Εικόνα 2.1.1: Λογότυπο Android

Η εξέλιξη του Android

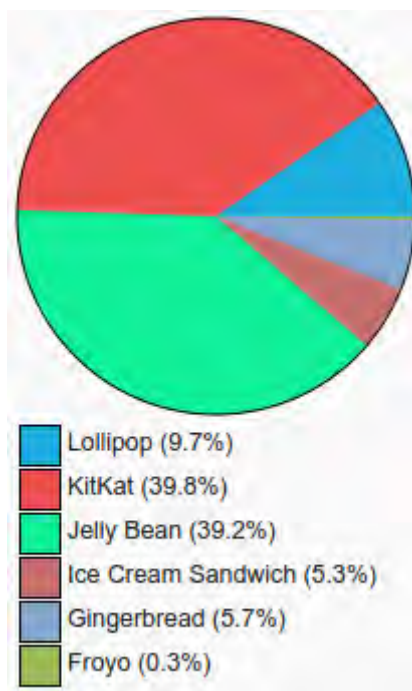
Η ιστορία εκδόσεων του λειτουργικού συστήματος Android των κινητών τηλεφώνων ξεκίνησε με την κυκλοφορία του Android beta, το Νοέμβριο του 2007^[6]. Η πρώτη εμπορική έκδοση ήταν το Android 1.0 που κυκλοφόρησε το Σεπτέμβριο του 2008. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance (OHA), και έχει πραγματοποιηθεί μια σειρά από ενημερώσεις στην λειτουργία του συστήματος από την αρχική του κυκλοφορία μέχρι σήμερα.

Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν θέμα από την ζαχαροπλαστική στην κωδική τους ονομασία και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1 που δεν τέθηκαν υπό συγκεκριμένα κωδικά ονόματα. Στον πίνακα που ακολουθεί παρουσιάζονται όλες οι εκδόσεις του Android μέχρι και σήμερα με το κωδικό όνομα, την ημερομηνία κυκλοφορίας και την έκδοση API τους. Στη στήλη που απεικονίζει τη διανομή, καθώς και στο γράφημα της εικόνας 2.1.2, παρουσιάζεται η διανομή εκδόσεων Android βασισμένη σε συσκευές που

επισκέφτηκαν το Play Store σε διάστημα 7 ημερών μέχρι τις 4 Μαΐου 2015. Εκδόσεις χωρίς ποσοστό διανομής αποτελούν αμελητέα ποσότητα.

Έκδοση	Κωδικό Όνομα	Ημερομηνία Κυκλοφορίας	API	Διανομή
1.0	Apple Pie	Σεπτέμβριος, 2008	1	-
1.1	Banana Bread	Φεβρουάριος, 2009	2	-
1.5	Cupcake	Απρίλιος, 2009	3	-
1.6	Donut	Σεπτέμβριος, 2009	4	-
2.0 – 2.1	Éclair	Οκτώβριος, 2009	5	-
2.2 – 2.2.3	Froyo	Μάιος, 2010	8	0,3%
2.3 – 2.3.7	Gingerbread	Δεκέμβριος, 2010	9	5,7%
3.0 – 3.2.6	Honeycomb	Φεβρουάριος, 2011	11	-
4.0 – 4.0.4	Ice Cream Sandwich	Οκτώβριος, 2011	14	5,3%
4.1 – 4.3.1	Jelly Bean	Ιούνιος, 2012	16	39,2%
4.4 – 4.4.4	KitKat	Οκτώβριος, 2013	19	39,8%
5.0 – 5.0.2	Lollipop	Νοέμβριος, 2014	21	9,7%

Πίνακας 2.1.1: Εκδόσεις Android



Εικόνα 2.1.2: Διανομή εκδόσεων Android

Εφαρμογές Android

Οι εφαρμογές (apps) που επεκτείνουν τη λειτουργικότητα των συσκευών, είναι γραμμένες κυρίως στη γλώσσα προγραμματισμού Java χρησιμοποιώντας το Android Software Development Kit (SDK). Το SDK περιλαμβάνει ένα πρόγραμμα εντοπισμού σφαλμάτων (debugger), βιβλιοθήκες λογισμικού, εξομοιωτή (emulator), documentation και tutorials. Η Google υποστήριξε ως ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) το Eclipse με τη χρήση του Android Development Tool (ADT) προσθέτου μέχρι το Δεκέμβριο του 2014, όπου και κυκλοφόρησε το Android Studio ως κύριο IDE για την ανάπτυξη εφαρμογών Android. Το Google Play Store είναι το κύριο «κατάστημα» εφαρμογών, εγκατεστημένο σε κάθε συσκευή Android που συμμορφώνεται με τις απαιτήσεις συμβατότητας της Google και έχει άδεια χρήσης του λογισμικού Google Mobile Services. Το Google Play Store επιτρέπει στους χρήστες να αναζητούν, να κατεβάζουν και να ενημερώνουν τις εφαρμογές που είναι δημοσιευμένες από την Google ή από άλλους προγραμματιστές. Κατά το πρώτο τετράμηνο του 2015 υπάρχουν στο Play Store παραπάνω από 1,5 εκατομμύριο εφαρμογές διαθέσιμες για κατέβασμα.

Υλικό

Η κύρια πλατφόρμα υλικού για το Android είναι η αρχιτεκτονική ARM (ARMv7 και ARMv8-A), καθώς και οι x86 και MIPS αρχιτεκτονικές (οι δύο τελευταίες κυρίως σε νεότερες εκδόσεις του Android). Από το Android 5.0 “Lollipop”, υποστηρίζονται 64-bit παραλλαγές από όλες τις πλατφόρμες. Επίσης από το 2012, έχουν κυκλοφορήσει στην αγορά Android συσκευές με Intel επεξεργαστές, όπως κινητά τηλέφωνα και ταμπλέτες (Lenovo K900, ASUS Padfone Mini). Ακόμη, από το Νοέμβριο του 2013 το Android 4.4 «συνιστά» τουλάχιστον 512 MB RAM.

Οι συσκευές Android ενσωματώνουν πολλά προαιρετικά εξαρτήματα υλικού (hardware) όπως βίντεο κάμερες, GPS, αισθητήρες προσανατολισμού, χειριστήρια παιχνιδιών, επιταχυνσιόμετρα, γυροσκόπια, βαρόμετρα, μαγνητόμετρα, αισθητήρες εγγύτητας, αισθητήρες πίεσης, θερμομέτρα, κ.α.

Αρχιτεκτονική

Το λειτουργικό σύστημα του Android είναι μία στοίβα από κομμάτια λογισμικού, τα οποία είναι διαιρεμένα σε 5 τμήματα και 4 κύρια στρώματα, όπως φαίνεται και στην εικόνα 2.1.3.



Εικόνα 2.1.3: Διάγραμμα της αρχιτεκτονικής του Android

- Linux kernel:** Η βάση της στοίβας είναι ο πυρήνας του Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται κυρίως στις εκδόσεις 3.4 ή 3.10 (Απρίλιος του 2014) του πυρήνα του Linux, οι οποίες υποστηρίζουν όλες τις κύριες λειτουργίες του συστήματος. Οι λειτουργίες αυτές αφορούν τη διαχείριση μνήμης, τη διαχείριση εργασιών, τις λειτουργίες δικτύου, την ασφάλεια του λειτουργικού, καθώς και ένα σύνολο από οδηγούς υλικού (hardware drivers). Συγκεκριμένα, οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του κομματιού του λογισμικού με το υλικό κομμάτι της συσκευής. Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Αυτό οφείλεται στο γεγονός ότι η Google έχει κάνει διάφορες αλλαγές, ώστε να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές.
- Libraries:** Στο δεύτερο επίπεδο υπάρχουν οι βιβλιοθήκες του Android. Αυτές ουσιαστικά αποτελούν τα APIs, τα οποία είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Αποτελούν ένα από τα δομικά υλικά των εφαρμογών, καθώς ενσωματώνονται και χρησιμοποιούνται για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++ και έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό.
- Dalvik Virtual Machine:** Σχεδόν όλο το σύνολο των APIs του Android βασίζεται στη γλώσσα προγραμματισμού Java. Στη Java ως γνωστόν υπάρχει η λεγόμενη Java εικονική μηχανή (Virtual Machine), στην οποία εκτελείται ο κώδικας bytecode των εφαρμογών. Στο Android υπάρχει αντίστοιχα η εικονική μηχανή Dalvik. Η

Dalvik είναι η εικονική μηχανή μέσω της οποίας τρέχουν οι εφαρμογές του Android. Η κάθε εφαρμογή τρέχει μέσω της δικής της εικονικής μηχανής στη δικιά της διεργασία και γι αυτό το λόγο καμιά εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται παράλληλα. Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java μεταγλωττίζονται (compile) σε αρχεία “.dex” για να τρέξουν στη VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου και χρόνου κατά την εκτέλεση. Το Android είναι από τη φύση του λειτουργικό σύστημα που υποστηρίζει την ταυτόχρονη εκτέλεση πολλών διεργασιών (multitasking) και έτσι επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα. Χάρη στο λιτό σχεδιασμό της Dalvik VM, αφού έχει σχεδιαστεί να έχει το ελάχιστο αντίκτυπο στη χρήση μνήμης, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

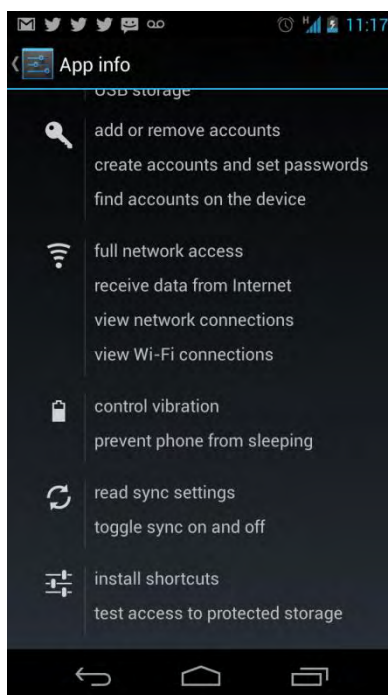
- **Android runtime:** Ο χρόνος εκτέλεσης των εφαρμογών του Android βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Εδώ βρίσκεται το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του χρόνου εκτέλεσης της Dalvik VM, τις λειτουργίες της οποίας περιγράψαμε παραπάνω.
- **Application framework:** Το Android παρέχει στους προγραμματιστές μια ανοικτού κώδικα πλατφόρμα ανάπτυξης, στην οποία μπορούν να δημιουργήσουν ιδιαίτερα καινοτόμες και πλούσιες σε υλικό εφαρμογές. Οι προγραμματιστές έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου και πάρα πολλές ακόμη δυνατότητες, οι οποίες βασίζονται στα APIs που είναι διαθέσιμα. Μερικές από τις δυνατότητες αυτές είναι:
 - Ένα σύνολο από γραφικά στοιχεία (Views) για τη δημιουργία του γραφικού περιβάλλοντος, όπως λιστών (List View), κειμένων (Text View), κουμπιών (Buttons), κ.α.
 - Ένας διαχειριστής περιεχομένου (Content Manager), ο οποίος επιτρέπει το διαμοιρασμό δεδομένων μεταξύ των εφαρμογών.
 - Ένας διαχειριστής πόρων (Resource Manager) για την πρόσβαση σε πόρους όπως αλφαριθμητικά (strings), εικόνες, κ.α.
 - Ένας διαχειριστής ειδοποιήσεων (Notification Manager), ο οποίος επιτρέπει την προβολή ειδοποιήσεων στη μπάρα κατάστασης (status bar).
 - Ένας διαχειριστής δραστηριοτήτων (Activity Manager), ο οποίος διαχειρίζεται τον κύκλο ζωής των εφαρμογών.
- **Applications:** Σε αυτό το επίπεδο βρίσκονται όλες οι εφαρμογές Android, οι οποίες είναι εγκατεστημένες σε μία συσκευή.

Ασφάλεια

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

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης.
- Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό (ID), ο οποίος είναι άγνωστος στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε εφαρμογή τρέχει στη δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες. Η κάθε εικονική μηχανή εκκινείται μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείται πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή.

Έτσι το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν παραπάνω. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στο χρήστη τη στιγμή της εγκατάστασης της και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να της παρέχει πρόσβαση στους πόρους που αυτή ζητάει. (Εικόνα 2.1.4)



Εικόνα 2.1.4: Δικαιώματα εφαρμογής κατά την εγκατάσταση της

2.2 XAMPP

Το **XAMPP**^{[7][8]} είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας, το οποίο περιέχει τον HTTP Apache server, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl. Είναι διαθέσιμο για Microsoft Windows, Linux, Solaris και Mac OS X.



Εικόνα 2.2.1: Λογότυπο XAMPP

Ετυμολογία

Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- **X**, αναφέρεται στο «cross-platform», που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας
- **A** Apache HTTP server
- **M** MySQL
- **P** PHP
- **P** Perl

Δυνατότητες και απαιτήσεις

Το XAMPP προϋποθέτει την ύπαρξη μόνο των λογισμικών συμπίεσης αρχείων zip, tar, 7z ή exe κατά την διάρκεια της εγκατάστασης. Έχει τη δυνατότητα αναβάθμισης σε νέες εκδόσεις του Apache server, της βάσης δεδομένων MySQL και των γλωσσών προγραμματισμού PHP και Perl. Συμπεριλαμβάνει επίσης τα πακέτα OpenSSL^[9] και το phpMyAdmin^[10].

Χρήση

Επίσημα, οι σχεδιαστές του XAMPP προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων χωρίς να είναι απαραίτητη η σύνδεση στο Internet. Στην πράξη όμως το XAMPP ορισμένες φορές χρησιμοποιείται για την φιλοξενία ιστοσελίδων στο Web. Όταν εγκατασταθεί στον τοπικό υπολογιστή, διαχειρίζεται τον localhost ως έναν απομακρυσμένο κόμβο.

Χαρακτηριστικά

Το XAMPP 5.6.8 περιλαμβάνει:

- Apache 2.4.12
- MySQL 5.6.24
- PHP 5.6.8
- phpMyAdmin 4.4.3
- OpenSSL 1.0.1m
- XAMPP Control Panel
- Mercury Mail Transport System 4.63
- FireZilla FTP Server 0.9.41
- Tomcat 7.0.56
- Strawberry Perl 7.0.56 Portable

2.3 Apache HTTP server

Ο **Apache HTTP server**^[11], γνωστός και απλά σαν Apache, είναι ένας εξυπηρέτης του παγκοσμίου ιστού. Όποτε ένας χρήστης επισκέπτεται έναν ιστότοπο, το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν εξυπηρέτη (server), μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρέτες ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, Linux και Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μία κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρέτη http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρέτες βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET.



Εικόνα 2.3.1: Λογότυπο Apache HTTP Server

2.4 PHP

Η **PHP**^{[12] [13] [14]} είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.



Εικόνα 2.4.1: Λογότυπο PHP

Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf, δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό κείμενο με όνομα `php.cgi` για προσωπική χρήση. Το κείμενο αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το συνδεδεμένο (online) βιογραφικό του σημείωμα. Αργότερα αυτό το κείμενο το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασισμένη αυτή τη φορά στη γλώσσα C και αριθμώνοντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασισμένοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0, η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6 αποσύρθηκαν λόγω της έλλειψης Unicode υποστήριξης. Η έκδοση PHP 7 αναμένεται να κυκλοφορήσει τον Οκτώβριο του 2015. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

2.5 MySQL

Η MySQL^[15] ^[16] είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα χρησιμοποιεί έναν εξυπηρετή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.



Εικόνα 2.5.1: Λογότυπο MySQL

Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, τη Google, το Facebook και το Twitter.

2.6 Το πρωτόκολλο επικοινωνίας HTTPS

Το HTTPS^[17] (Hypertext Transfer Protocol Secure) χρησιμοποιείται στην πληροφορική για να δηλώσει μία ασφαλή σύνδεση http. Ένας σύνδεσμος (URL) που αρχίζει με το πρόθεμα https υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα (443 αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα. Το σύστημα αυτό σχεδιάστηκε αρχικά από την εταιρία Netscape Communications Corporation, για να χρησιμοποιηθεί σε ιστότοπους, όπου απαιτείται αυθεντικοποίηση (authentication) χρηστών και κρυπτογραφημένη επικοινωνία. Σήμερα χρησιμοποιείται ευρέως στο διαδίκτυο, όπου χρειάζεται αυξημένη ασφάλεια, διότι διακινούνται ευαίσθητες πληροφορίες (πχ αριθμοί πιστωτικών καρτών, passwords κοκ).

Τρόπος λειτουργίας

Το HTTPS δεν είναι ξεχωριστό πρωτόκολλο, όπως εσφαλμένως μερικοί πιστεύουν, αλλά αναφέρεται στον συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL) / Transport Layer Security (TLS)^[18]. Η κρυπτογράφηση που χρησιμοποιείται διασφαλίζει ότι τα κρυπτογραφημένα δεδομένα δεν θα μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες ή από επιθέσεις man-in-the-middle.

Για να χρησιμοποιηθεί το HTTPS σε έναν εξυπηρετή, θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Σε εξυπηρετές που χρησιμο-

ποιούν το λειτουργικό σύστημα UNIX αυτό μπορεί να γίνει μέσω του προγράμματος OpenSSL. Στην συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία πιστοποιεί ότι ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτό οι χρήστες μπορούν να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει.

2.7 Wireshark

Το **Wireshark**^{[19] [20]} είναι ένα ελεύθερο και ανοιχτού κώδικα λογισμικό ανάλυσης πρωτοκόλλων δικτύου υπολογιστών. Χρησιμοποιείται για την ανάλυση του δικτύου, την παρακολούθηση του δικτύου, τον εντοπισμό και την αντιμετώπιση προβλημάτων στα δίκτυα και για την εκπαίδευση. Το αρχικό όνομα του προγράμματος ήταν Ethereal, και τον Μάιο του 2006 άλλαξε σε Wireshark για λόγους εμπορικών σημάτων.

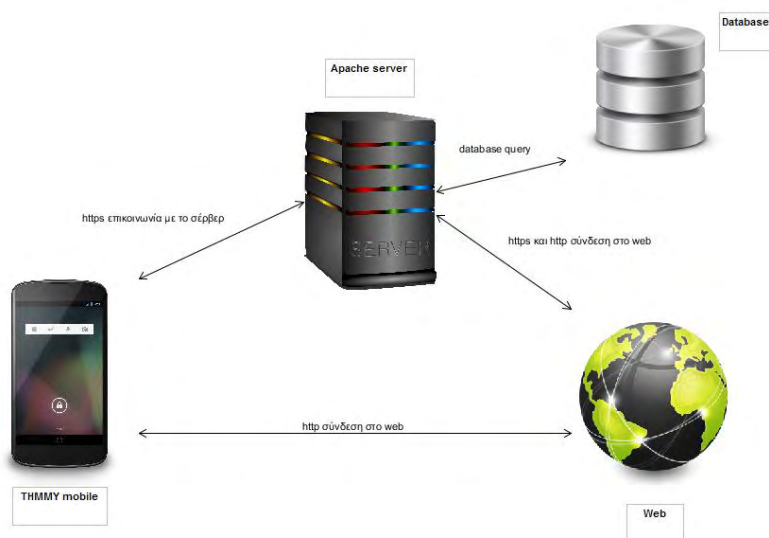
Είναι διαθέσιμο για όλα τα κύρια λειτουργικά συστήματα όπως Windows, Linux και Mac OS X. Χρησιμοποιεί το GTK+ για το γραφικό περιβάλλον και το Pcap για τη σύλληψη πακέτων. Διατίθεται υπό τους όρους της Γενικής Άδειας δημόσιας χρήσης GNU.



Εικόνα 2.7.1: Λογότυπο Wireshark

3 Αρχιτεκτονική συστήματος

3.1 Σχέδιο υλοποίησης



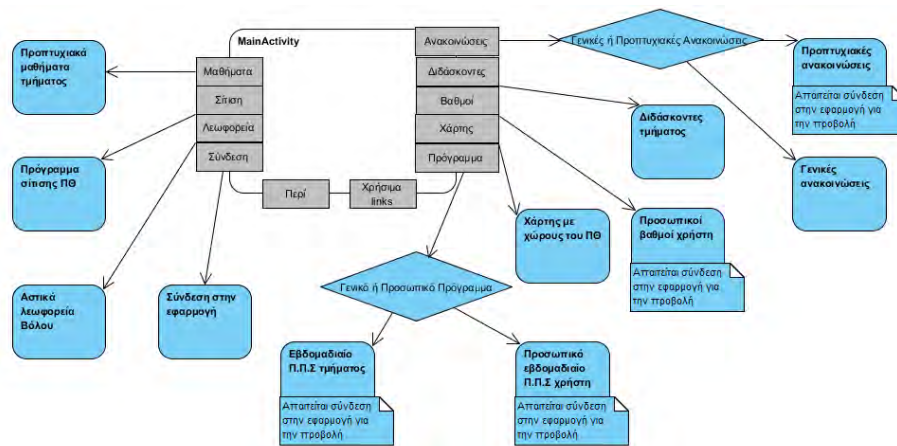
Εικόνα 3.1.1: Παρουσίαση των λειτουργικών μονάδων της εργασίας

Στην εικόνα 3.1.1 παρουσιάζεται η γενική δομή των μονάδων της παρούσας διπλωματικής εργασίας. Η περάτωσή της αποτελείται από δύο βασικά κομμάτια, την υλοποίηση της εφαρμογής και την υλοποίηση του εξυπηρετή.

- Η mobile εφαρμογή αναπτύχθηκε στο λειτουργικό σύστημα Android. Είναι μία native εφαρμογή γραμμένη σε κώδικα Java. Η κύρια λειτουργικότητά της είναι η ανάλυση (parse) δεδομένων διάφορων ιστοσελίδων και η παρουσίασή τους στην οθόνη του κινητού τηλεφώνου του χρήστη. Χρησιμοποιεί ασφαλή επικοινωνία (https) με έναν Apache εξυπηρετή (server) για ανταλλαγή ευαίσθητων δεδομένων. Επίσης, σε κάποια σημεία, όπου δεν υπάρχει θέμα ασφάλειας δεδομένων, χρησιμοποιεί απευθείας σύνδεση στο διαδίκτυο.
- Από την άλλη πλευρά, ο εξυπηρετής υλοποιήθηκε με την βοήθεια του ελεύθερου λογισμικού XAMPP. Η κύρια λειτουργικότητά του είναι να συνδέεται σε ιστοσελίδες στο διαδίκτυο, είτε με τη χρήση username και password (https σύνδεση) είτε χωρίς (http σύνδεση), και να επιστρέφει στην εφαρμογή τα περιεχόμενα των ιστοσελίδων που χρειάζεται η τελευταία. Ακόμα, χρησιμοποιεί και μια μικρή βάση δεδομένων, στην οποία είναι αποθηκευμένες διάφορες πληροφορίες.

3.2 Ανάλυση υπομονάδων

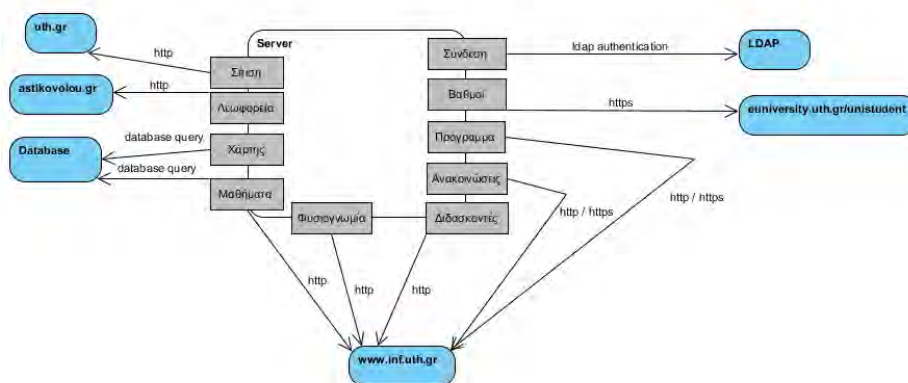
Android εφαρμογή



Εικόνα 3.2.1: Παρουσίαση της δομής της Android εφαρμογής

Η εφαρμογή αναφέρεται στους προπτυχιακούς φοιτητές του ΤΗΜΜΥ του ΠΘ και συνεπώς ακολουθεί τους κανόνες που ισχύουν στο τμήμα. Η εφαρμογή είναι ελεύθερη για χρήση από οποιονδήποτε το επιθυμεί. Για την προσπέλαση όμως συγκεκριμένων πληροφοριών, όπως το πρόγραμμα, οι ανακοινώσεις προπτυχιακών και οι προσωπικοί βαθμοί, απαιτείται η σύνδεση του χρήστη στην εφαρμογή με τους παρεχόμενους από τη σχολή κωδικούς.

Εξυπηρέτης (Server)



Εικόνα 3.2.2: Παρουσίαση της δομής του Apache server

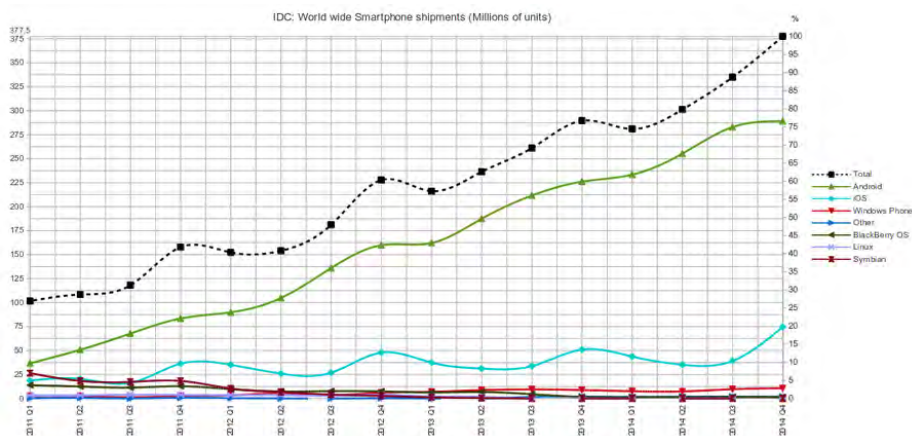
Ο εξυπηρέτης αποτελεί το ενδιάμεσο στάδιο μεταξύ της εφαρμογής και του διαδικτύου. Ορισμένες από τις ενέργειες που πραγματοποιούνται είναι:

- Idar authentication^{[21] [22]}, για την επιβεβαίωση ότι ο χρήστης αποτελεί φοιτητή του τμήματος (κρυπτογραφημένη επικοινωνία).
- σύνδεση στο studentWeb της σχολής και επιστροφή των βαθμών καθώς και της δήλωσης του χρήστη (κρυπτογραφημένη επικοινωνία).
- σύνδεση στο site της σχολής και επιστροφή του προγράμματος προπτυχιακών σπουδών καθώς και των ανακοινώσεων προπτυχιακών (κρυπτογραφημένη επικοινωνία).
- επιστροφή περιεχομένων διάφορων ιστοσελίδων, όπως τους διδάσκοντες του τμήματος, το πρόγραμμα σίτισης, κ.α. (μη κρυπτογραφημένη επικοινωνία).
- χρήση βάσης δεδομένων MySQL.

3.3 Λογική σχεδιασμού

Γιατί Android;

Η επιλογή του λειτουργικού συστήματος Android ήταν μία συνειδητή επιλογή, καθώς είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές iOS και Windows Phone μαζί, όπως φαίνεται και στην εικόνα 3.3.1 που ακολουθεί.



Εικόνα 3.3.1: Πωλήσεις έξυπνων τηλεφώνων παγκοσμίως μέχρι το τέλος του 2014^[23]

Γιατί εξυπηρέτης;

Η υιοθέτηση της λογικής του εξυπηρέτη οφείλεται σε πολλούς λόγους, καθένας σημαντικός για την λειτουργία της εφαρμογής.

1. «Προσπέλαση» **απαίτησης για VPN**. Ο εξυπηρετής της εφαρμογής έχει υλοποιηθεί μέσα στο δίκτυο του τμήματος. Με αυτόν τον τρόπο δεν απαιτείται σύνδεση μέσω VPN για τη προσπέλαση των ηλεκτρονικών υπηρεσιών της Γραμματείας του τμήματος. Έτσι ο φοιτητής μπορεί να ελέγξει τις βαθμολογίες των μαθημάτων του εύκολα και γρήγορα χωρίς την χρήση επιπλέον προγραμμάτων.
2. **Δυναμικός έλεγχος των υπερσυνδέσεων των σελίδων (url) μέσω του εξυπηρετή**. Αυτό σημαίνει ότι εάν κάποια στιγμή αλλάξει κάποιος υπερσύνδεσμος, δεν θα απαιτηθεί αναβάθμιση της εφαρμογής, καθώς το πρόβλημα λύνεται τοπικά στον εξυπηρετή.
3. **Ανάλυση των περιεχομένων των σελίδων και αποστολή τους στην εφαρμογή**. Στο μεγαλύτερο κομμάτι του κώδικα του εξυπηρετή πραγματοποιείται επεξεργασία των περιεχομένων των σελίδων που απαιτούνται. Έτσι στην εφαρμογή επιστρέφονται μόνο τα περιεχόμενα που αυτή χρειάζεται.
4. **Υλοποίηση βάσης δεδομένων στον εξυπηρετή**. Η βάση δεδομένων εξυπηρετεί στην αποθήκευση των page ids των προπτυχιακών μαθημάτων. Κάτι τέτοιο κρίθηκε απαραίτητο, επειδή δεν υπήρχε τρόπος ανάλυσής τους (parse) από κάποια σελίδα στον ιστότοπο της σχολής. Επίσης η βάση φιλοξενεί διάφορες συντεταγμένες σημείων για την εύρεσή τους πάνω στο χάρτη.

Ασφάλεια επικοινωνίας με τον εξυπηρετή

Η μεταφορά κατά την επικοινωνία με τον εξυπηρετή ευαίσθητων προσωπικών δεδομένων (username, password, προσωπικοί βαθμοί) οδήγησε στην κρυπτογράφηση ολόκληρης της επικοινωνίας μέσω του πρωτοκόλλου HTTPS^[24]. Για την επιβεβαίωση ότι η επικοινωνία με τον εξυπηρετή είναι πλέον κρυπτογραφημένη χρησιμοποιήθηκε το Wireshark, με το οποίο καταγράφηκαν τα πακέτα που αντάλλαξαν η εφαρμογή με τον εξυπηρετή (Εικόνα 3.3.2). Όπως φαίνεται στην παρακάτω εικόνα, όλη η επικοινωνία είναι κρυπτογραφημένη υπό το πρωτόκολλο TLS.

Protocol	Length	Info
TCP	74	56570->443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=6194463 TSecr=0 wS=128
TCP	74	443->56570 [SYN, ACK] Seq=0 Ack=1 Wlen=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1424344 TSecr=6194463 wS=128
TCP	66	56570->443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=6194463 TSecr=1424344
TCP	74	56571->443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=6194463 TSecr=0 wS=128
TCP	74	443->56571 [SYN, ACK] Seq=0 Ack=1 Wlen=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1424344 TSecr=6194463 wS=128
TCP	66	56571->443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=6194463 TSecr=1424344
TLSv1.2	260	Client Hello
TCP	66	443->56570 [ACK] Seq=1 Ack=195 Win=30080 Len=0 TSval=1424348 TSecr=6194466
TLSv1.2	260	Client Hello
TLSv1.2	951	Server Hello, Certificate, Server Key Exchange, Server Hello Done
TCP	66	56570->443 [ACK] Seq=195 Ack=886 Win=32128 Len=0 TSval=6194467 TSecr=1424348
TCP	66	443->56571 [ACK] Seq=1 Ack=195 Win=30080 Len=0 TSval=1424348 TSecr=6194467
TLSv1.2	951	Server Hello, Certificate, server key exchange, server hello done
TCP	66	56571->443 [ACK] Seq=195 Ack=886 Win=32128 Len=0 TSval=6194467 TSecr=1424348
TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
TLSv1.2	286	Application Data
TLSv1.2	283	Application Data
TCP	66	443->56571 [ACK] Seq=937 Ack=538 Win=31104 Len=0 TSval=1424361 TSecr=6194470
TCP	66	443->56570 [ACK] Seq=937 Ack=541 Win=31104 Len=0 TSval=1424361 TSecr=6194469
TLSv1.2	1514	Application Data, Application Data
TCP	1514	[TCP segment of a reassembled PDU]
TCP	66	56570->443 [ACK] Seq=541 Ack=3833 Win=37888 Len=0 TSval=6194832 TSecr=1424713
TCP	1514	[TCP segment of a reassembled PDU]
TCP	1514	[TCP segment of a reassembled PDU]
TCP	66	56570->443 [ACK] Seq=541 Ack=6729 Win=43776 Len=0 TSval=6194832 TSecr=1424713
TCP	1514	[TCP segment of a reassembled PDU]
TCP	66	56570->443 [ACK] Seq=541 Ack=9625 Win=49536 Len=0 TSval=6194832 TSecr=1424713
TCP	1514	[TCP segment of a reassembled PDU]
TCP	1514	[TCP segment of a reassembled PDU]
TCP	66	56570->443 [ACK] Seq=541 Ack=12521 Win=55296 Len=0 TSval=6194832 TSecr=1424713
TCP	1514	[TCP segment of a reassembled PDU]

Εικόνα 3.3.2: Επικοινωνία εφαρμογής – server

4 Ανάλυση υπομονάδων συστήματος

4.1 Android εφαρμογή

Πριν την περιγραφή του κώδικα της εφαρμογής που υλοποιήθηκε, κρίνεται σκόπιμο να αναφερθούν ορισμένες έννοιες που θα χρησιμοποιηθούν στη συνέχεια^[25].

- **Activity:** Η κλάση Activity παρέχει μία οθόνη αλληλεπίδρασης με το χρήστη μέσω της μεθόδου `setContentView(View)`. Η μέθοδος αυτή καλείται μέσα στην συνάρτηση `onCreate()`, η οποία αρχικοποιεί το activity μας. Στην εφαρμογή όλες οι κύριες επιλογές του χρήστη έχουν υλοποιηθεί ως Activities.
- **Fragment:** Είναι ένα κομμάτι από μία διεπαφή χρήστη (UI) μιας εφαρμογής, το οποίο τοποθετείται σε ένα Activity και εξαρτάται άμεσα από αυτό. Τα fragments χρησιμοποιούνται, όταν θέλουμε να αξιοποιήσουμε πολλές διεπαφές χρήστη μέσα σε ένα Activity.
- **FragmentManager:** Διεπαφή για την αλληλεπίδραση με τα fragments μέσα σε ένα activity.
- **AsyncTask:** Επιτρέπει μια σωστή και εύκολη χρήση του νήματος της διεπαφής χρήστη (UI thread). Αυτή η κλάση επιτρέπει την πραγματοποίηση λειτουργιών παρασκήνιου και δημοσιοποίηση των αποτελεσμάτων στο νήμα της διεπαφής χρήστη. Όλες οι λειτουργίες δικτύου οφείλουν να υλοποιηθούν με ασύγχρονη διεργασία (AsyncTask).
- **SharedPreferences:** Ένας από τους πολλούς τρόπους που προσφέρει το Android για την αποθήκευση δεδομένων της εφαρμογής. Με την ετικέτα “MODE_PRIVATE” κατά τη δημιουργία των `sharedpreferences` δηλώνεται ότι τα περιεχόμενα του αρχείου που δημιουργείται είναι προσπελάσιμα μόνο από την ίδια την εφαρμογή (ή άλλες εφαρμογές που μοιράζονται το ίδιο user ID).
- **WebView:** Είναι μία προβολή (view) που παρουσιάζει web σελίδες.
- **ListView:** Είναι μία ομαδική προβολή (view group), η οποία αναπαριστά μία λίστα από αντικείμενα με δυνατότητα κύλισης. Τα αντικείμενα εισέρχονται αυτόματα στη λίστα μέσω ενός αντάπτορα (Adapter).
- **ExpandableListView:** Είναι μια προβολή, η οποία παρουσιάζει αντικείμενα σε μία κάθετη δύο-επιπέδων λίστα. Η διαφορά με το `ListView` είναι ότι επιτρέπει δύο επίπεδα. Οι ομάδες μπορούν ανεξάρτητα να επεκταθούν, ώστε να αναπαραστήσουν τα αντικείμενά τους.
- **TableLayout:** Είναι μια διάταξη, η οποία κατανέμει τα αντικείμενά της σε γραμμές και στήλες.
- **DrawerLayout:** Επιτρέπει “drawer” προβολές να εμφανίζονται από τη γωνία του παραθύρου.
- **TextView:** Παρουσιάζει το κείμενο στο χρήστη.
- **ViewPager:** Είναι μια διάταξη που επιτρέπει στο χρήστη να γυρίζει αριστερά και δεξιά τις σελίδες περιεχομένων.
- **Intent:** Είναι μια περιληπτική περιγραφή μιας ενέργειας. Μπορεί να χρησιμοποιηθεί μαζί με το “`startActivity`” για να εκκινήσει ένα νέο Activity ή μαζί με το “`startService(Intent)`” για να επικοινωνήσει με ένα Service παρασκήνιου.

- **Jsoup**^[26]: Είναι μια Java βιβλιοθήκη για επεξεργασία HTML. Παρέχει ένα πολύ βολικό API για την εξαγωγή και το χειρισμό των δεδομένων.
- **Volley**^[27]: Είναι μια HTTP βιβλιοθήκη, η οποία κάνει τη δικτύωση (networking) για τις Android εφαρμογές πιο εύκολη και κυρίως πιο γρήγορη.
- **JSON**^[28]: (**J**ava**S**cript **O**bject **N**otation) Είναι μία ανεξαρτήτου γλώσσας ανοιχτού προτύπου μορφή (open standard format) που χρησιμοποιεί αναγνωρίσιμο από τον άνθρωπο (human readable) κείμενο για να μεταδώσει δεδομένα-αντικείμενα. Χρησιμοποιείται συνήθως για να μεταδώσει δεδομένα μεταξύ ενός εξυπηρετή και μιας εφαρμογής ως ένα εναλλακτικό XML. Με το σύμβολο “{” ορίζεται ένα JSON αντικείμενο, ενώ με το σύμβολο “[” ορίζεται ένας JSON πίνακας.

```

{
  "employees": [
    {
      "firstName": "John",
      "lastName": "Doe"
    },
    {
      "firstName": "Anna",
      "lastName": "Smith"
    },
    {
      "firstName": "Peter",
      "lastName": "Jones"
    }
  ]
}

```

Εικόνα 4.1.1: Παράδειγμα JSON

- **SpannableString**: Είναι μία κλάση για κείμενο, το οποίο είναι αμετάβλητο και στο οποίο μπορούν να προσκολληθούν αντικείμενα (click event).
- **ClickableSpan**: Είναι μία κλάση με αντικείμενα, τα οποία μπορούν να προσκολληθούν σε ένα κείμενο ενός TextView με μία μέθοδο κίνησης όπως η LinkMovementMethod.
- **ArrayList**: Είναι μια εφαρμογή της κλάσης List, η οποία υποστηρίζεται από έναν πίνακα.
- **HashMap**: Αντιστοιχεί σε μία εφαρμογή της κλάσης Map. Είναι μια δομή δεδομένων που αποτελείται από ένα σετ κλειδιών και τιμών, στο οποίο κάθε κλειδί αντιστοιχεί σε μια τιμή.

Γενική περιγραφή

Υλοποίηση του πελάτη (client) για την επικοινωνία με τον εξυπηρετή

Ο κώδικας 4.1.1 παρουσιάζει την αρχικοποίηση του πελάτη. Η μέθοδος αυτή υλοποιείται σε όλα τα activities, όπου υπάρχει επικοινωνία με τον εξυπηρετή. Χρησιμοποιείται η μέθοδος HTTP POST, ενώ οι παράμετροι (tag, κ.α.) που αποστέλλονται κατά την επικοινωνία δηλώνονται κατά τη δημιουργία του constructor της κλάσης BasicNameValuePair. Το αποτέλεσμα της επικοινωνίας επιστρέφεται σαν αλφαριθμητικό (string). Όπως αναφέραμε και στο κεφάλαιο 3.1, η επικοινωνία αυτή είναι πλήρως κρυπτογραφημένη. Στον κώδικα 4.1.2 παρουσιάζεται ο τρόπος, με τον οποίο επιτυγχάνεται η κρυπτογράφηση.

```

1. DefaultHttpClient client = (DefaultHttpClient) WebClientDevWrapper.
2.     getNewHttpClient(getApplicationContext());
3. HttpPost post = new HttpPost(url);
4. ArrayList localArrayList = new ArrayList(0);
5. BasicNameValuePair localBasicNameValuePair = new BasicNameValuePair
6.     ("tag", "tag_name");
7. localArrayList.add(localBasicNameValuePair);
8. UrlEncodedFormEntity localUrlEncodedFormEntity = new UrlEncodedFormEntity
9.     (localArrayList);
10. post.setEntity(localUrlEncodedFormEntity);
11. String response = EntityUtils.toString(client.execute(post).getEntity(), "
12.     UTF-8");

```

Κώδικας 4.1.1: Αρχικοποίηση client και εισαγωγή παραμέτρων

```

1. CertificateFactory cf = CertificateFactory.getInstance("X.509");
2. InputStream caInput = new BufferedInputStream (con-
3.     text.getAssets().open("server.crt"));
4. Certificate ca = cf.generateCertificate(caInput);
5.
6. KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
7. trustStore.load(null, null);
8. trustStore.setCertificateEntry("ca", ca);
9.
10. SSLSocketFactory sf = new MySSLSocketFactory(trustStore);
11. sf.setHostnameVerifier(SSLSocketFactory.STRICT_HOSTNAME_VERIFIER);
12.
13. HttpParams params = new BasicHttpParams();
14. HttpProtocolParams.setVersion(params, HttpVersion.HTTP_1_1);
15. HttpProtocolParams.setContentCharset(params, HTTP.UTF_8);
16.
17. // Set the timeout in milliseconds until a connection is established.
18. // The default value is zero, that means the timeout is not used.
19. int timeoutConnection = 5000;
20. HttpConnectionParams.setConnectionTimeout(params, timeoutConnection);
21. // Set the default socket timeout (SO_TIMEOUT)
22. // in milliseconds which is the timeout for waiting for data.
23. int timeoutSocket = 12000;
24. HttpConnectionParams.setSoTimeout(params, timeoutSocket);
25.
26. SchemeRegistry registry = new SchemeRegistry();
27. registry.register(new Scheme("http", PlainSocketFactory.getSocketFactory(),
28.     80));
29. registry.register(new Scheme("https", sf, 443));
30.
31. ClientConnectionManager ccm = new ThreadSafeClientConnManager(params, reg-
32.     ister ccm = new ThreadSafeClientConnManager(params, registry);

```

Κώδικας 4.1.2: Δημιουργία του client για την επικοινωνία με το server

(Κώδικας 4.1.2)^[29] Ο πελάτης χρησιμοποιεί το ίδιο πιστοποιητικό (certificate) που χρησιμοποιεί και ο εξυπηρετής για την επικοινωνία, το οποίο έχει τοποθετηθεί στο φάκελο assets της εφαρμογής. Η κλάση SSLSocketFactory επικυρώνει τον HTTPS εξυπηρετή με τη χρήση του TLS πρωτοκόλλου και σύμφωνα με το πιστοποιητικό που έχει δηλωθεί. Η επικοινωνία θα απορριφθεί κατά τη διάρκεια του SSL session handshake, αν ο HTTPS εξυπηρετής-στόχος χρησιμοποιεί διαφορετικό πιστοποιητικό. Επίσης, οι παράμετροι setConnectionTimeout και setSoTimeout χρησιμεύουν ως timeouts σε περίπτωση που υπάρξει κάποιο πρόβλημα κατά την επικοινωνία. Τέλος, δηλώνεται ότι για την https επικοινωνία θα χρησιμοποιηθεί η πόρτα 443.

Επικοινωνία μεταξύ των activities

Η επικοινωνία και η μεταφορά παραμέτρων μεταξύ των activities πραγματοποιείται με τη βοήθεια της κλάσης Intent (Κώδικας 4.1.3). Με τη μέθοδο putExtra εισάγονται οι παράμετροι που θα αποσταλούν, ενώ με τη μέθοδο getStringExtra λαμβάνονται οι παράμετροι που έχουν σταλεί.

```
1. Intent i = new Intent(Context package Context, Class<?> cls);
2. i.putExtra(String name, String value);
3. startActivity(i);
4.
5. Intent i = getIntent();
6. String name = i.getStringExtra("string_name");
```

Κώδικας 4.1.3: Επικοινωνία μεταξύ των activities

Activity animation

Για την αλλαγή της απεικόνισης (animation) κατά τη μετάβαση από ένα activity σε ένα άλλο χρησιμοποιείται η μέθοδος “overridePendingTransition” (Κώδικας 4.1.4). Η πρώτη παράμετρος αναφέρεται στην απεικόνιση εξόδου, ενώ η δεύτερη στην απεικόνιση εισόδου.

```
1. overridePendingTransition(int exitAnim, int enterAnim)
```

Κώδικας 4.1.4: Αλλαγή στο animation

Ανανέωση περιεχομένων σελίδων

Σε όλα τα activities, στα οποία γίνεται φόρτωση των περιεχομένων από τα sharedPreferences της εφαρμογής, δίνεται η δυνατότητα ανανέωσης των περιεχομένων και αποθήκευσής τους ξανά στα sharedPreferences με την αντίστοιχη ετικέτα.

```

1. Button btn_reload = (ImageButton)findViewById(R.id.sub_img2);
2. btn_reload.setOnClickListener(new View.OnClickListener() {
3.
4.     @Override
5.     public void onClick(View view) {
6.         // clear listview
7.         setListAdapter(null);
8.         new NewConnection().execute(prefs.getString("username", null), pr
9.         efs.getString("password", null));
10.    }
11. });
12. private SharedPreferences prefs;
13. private SharedPreferences.Editor editor;
14.
15. editor = prefs.edit();
16. editor.putString("ann", str);
17. editor.commit();

```

Κώδικας 4.1.5: Ανανέωση των περιεχομένων των ανακοινώσεων

Περιγραφή των activities

MainActivity.java

Αποτελεί την κύρια κλάση της εφαρμογής. Αρχικά κατά την εκκίνηση της εφαρμογής, ελέγχεται η κατάσταση του δικτύου στη συσκευή με τη χρήση της κλάσης `ConnectivityManager`. Σε περίπτωση που δεν είναι ενεργοποιημένα τα δεδομένα δικτύου, εμφανίζεται μήνυμα για την ενεργοποίησή τους (Κώδικας 4.1.6). Αυτό κρίθηκε σκόπιμο, καθώς η εφαρμογή χρησιμοποιεί επικοινωνία με το διαδίκτυο, ώστε να προσκομίσει και στη συνέχεια να παρουσιάσει τα δεδομένα της.

```

1. if(cm.getActiveNetworkInfo() == null || !cm.getActiveNetworkInfo().isAvail
2.     able()
3.     || !cm.getActiveNetworkInfo().isConnected()){
4.     AlertDialog
5.     log.Builder alertDialogBuilder = new AlertDialog.Builder(
6.     MainActivity.this);}

```

Κώδικας 4.1.6: Έλεγχος για δεδομένα στη συσκευή

Αφού επιβεβαιωθεί ότι τα δεδομένα της συσκευής είναι ενεργοποιημένα, δημιουργούνται 5 παράλληλοι πελάτες (clients) με τη χρήση 5 διαφορετικών νημάτων (threads) για την προσκόμιση των περιεχομένων της συσκευής που δεν χρειάζονται σύνδεση από το χρήστη. Εδώ αξίζει να τονιστεί ότι ενεργοποιείται ένα barrier, το

οποίο εγγυάται ότι όλα τα νήματα έχουν τελειώσει τις εργασίες τους. Αυτό επιτυγχάνεται με τη χρήση της κλάσης `CountDownLatch` και του «μετρητή» `latch`, τον οποίο μειώνει κάθε νήμα όταν τελειώνει την εργασία του. Η ιδέα είναι ότι ο μετρητής αρχικοποιείται σύμφωνα με τον αριθμό των νημάτων που χρησιμοποιούνται και εμείς αναμένουμε έως ότου γίνει μηδέν. Τέλος, τα επιστρεφόμενα αλφαριθμητικά από κάθε επικοινωνία αποθηκεύονται στα `sharedpreferences` της εφαρμογής με τη αντίστοιχη ετικέτα (`tag`). Μετά το πέρας αυτής της διαδικασίας, μπορεί να πραγματοποιηθεί η προσπέλαση οποιουδήποτε `activity`.

```
1. DidaskThread thread1 = new DidaskThread();
2. thread1.start();
3. private class DidaskThread extends Thread{
4.     public void run(){
5.         try {
6.             didask = requestDidask(POST_URL);
7.             latch.countDown();
8.         } catch (IOException e){
9.             }
10.    }
11. }
```

Κώδικας 4.1.7: Δημιουργία νήματος για προσκόμιση των διδασκόντων του τμήματος

Login.java

Η κλάση αυτή αποτελεί το `activity`, στο οποίο λαμβάνει χώρα η σύνδεση του χρήστη στην εφαρμογή για την προσπέλαση όλων των περιεχομένων της, ή η αποσύνδεση από αυτήν, εάν είναι ήδη συνδεδεμένος.

Στην πρώτη περίπτωση, με την συμπλήρωση όλων των πεδίων και το πάτημα του κουμπιού «Σύνδεση», δημιουργούνται δύο νήματα. Το ένα είναι υπεύθυνο για την προσκόμιση των περιεχομένων που χρειάζονται σύνδεση από τον ιστότοπο της σχολής και το άλλο για την προσκόμιση των προσωπικών βαθμών του χρήστη από το `studentWeb` της σχολής. Απαραίτητη προϋπόθεση για την άρτια προσκόμιση των περιεχομένων των σελίδων αυτών είναι τα στοιχεία του χρήστη να είναι έγκυρα. Στην περίπτωση αυτή τα στοιχεία χρήστη αποθηκεύονται στα `sharedpreferences` της εφαρμογής, όπως επίσης αποθηκεύονται και τα περιεχόμενα των ιστοσελίδων που επιστράφηκαν (Κώδικας 4.1.8). Η ετικέτα “`logged`” χρησιμοποιείται για τον έλεγχο της συνδεσιμότητας του χρήστη στην εφαρμογή. Μετά από ένα δευτερόλεπτο η εφαρμογή ανακατευθύνεται στην αρχική σελίδα (Κώδικας 4.1.9). Η σημείωση (`flag`) “`FLAG_TOP_CLER_TOP`” σβήνει όλα τα παλιά `activities` από τη στοίβα και θέτει το `MainActivity` στην κορυφή της. Αντιθέτως, αν τα στοιχεία δεν είναι έγκυρα, εκτυπώνεται μήνυμα σφάλματος στην οθόνη.

Στην δεύτερη περίπτωση, πατώντας πάνω στο κουμπί «Αποσύνδεση» πραγματοποιείται αποσύνδεση του χρήστη από την εφαρμογή και ταυτόχρονα σβήνονται όλα τα στοιχεία του από τη συσκευή (Κώδικας 4.1.10).

```

1. editor = prefs.edit();
2. editor.clear();
3. editor.putString("username", USERNAME);
4. editor.putString("password", PASSWORD);
5. editor.putString("logged", "true");
6. editor.putString("ann", ann_str);
7. editor.putString("sch", sch_str);
8. editor.putString("mark", grad);
9. editor.commit();

```

Κώδικας 4.1.8: Αποθήκευση προσωπικών στοιχείων χρήστη και περιεχομένων ιστοσελίδων στα sharedpreferences της εφαρμογής

```

1. int timeout = 1000; // make the activity visible for 1 second
2. Timer timer = new Timer();
3. timer.schedule(new TimerTask() {
4.
5.     @Override
6.     public void run() {
7.         finish();
8.         Intent homepage = new Intent(Login.this, MainActivity.class);
9.         homepage.putExtra("back", "true");
10.        homepage.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
11.        startActivity(homepage);
12.        finish();
13.    }
14. }, timeout);

```

Κώδικας 4.1.9: Ανακατεύθυνση στην αρχική σελίδα

```

1. editor = prefs.edit();
2. editor.clear();
3. editor.commit();

```

Κώδικας 4.1.10: Αποσύνδεση από την εφαρμογή

AnakoinoseisStep.java

Το activity αυτό αποτελεί ένα ενδιάμεσο βήμα, στο οποίο επιλέγονται είτε οι γενικές ανακοινώσεις της σχολής, είτε οι ανακοινώσεις προπτυχιακών.

ProptixiakesAnakoinoseis.java

Η κλάση αυτή χειρίζεται τις ανακοινώσεις προπτυχιακών. Απαραίτητη προϋπόθεση είναι η σύνδεση του χρήστη με τους προσωπικούς του κωδικούς. Για το λόγο αυτό κατά την εκκίνηση του συγκεκριμένου activity πραγματοποιείται έλεγχος συνδεσιμότητας, όπως φαίνεται στον κώδικα 4.1.11.


```

1. if (prefs.getString("logged", "false").equals("true")){
2.     listView = getListView();
3.     String str = prefs.getString("ann", "false");
4.     new Parser().execute(str);
5. }else{
6.     Toast.makeText(ProptixiakiesAnakoynoseis.this, " Παρακαλώ τοποθετήστε
    τα στοιχεία σας!", Toast.LENGTH_LONG).show();
7. }

```

Κώδικας 4.1.11: Έλεγχος για το αν ο χρήστης είναι συνδεδεμένος

Αν ο χρήστης είναι συνδεδεμένος στην εφαρμογή, φορτώνονται από τα `sharedpreferences` τα περιεχόμενα των προπτυχιακών ανακοινώσεων της σχολής, τα οποία ελήφθησαν κατά την είσοδο του στην εφαρμογή (`Login.java`). Οι ανακοινώσεις στον ιστότοπο της σχολής βρίσκονται σε μορφή RSS και υποστηρίζουν τη γλώσσα σήμανσης XML. Η μορφή των RSS δίνεται στον κώδικα 4.1.12. Κάθε ανακοίνωση βρίσκεται μέσα στην ετικέτα “item”, όπου περιέχονται διάφορες σημαντικές πληροφορίες όπως ο τίτλος της ανακοίνωσης, η περιγραφή της, ο σύνδεσμός της και η μέρα δημοσιοποίησής της.

```

1. <?xml version="1.0" encoding="UTF-8" ?>
2. <rss version="2.0">
3. <channel>
4. <title>RSS Title</title>
5. <description>This is an example of an RSS feed</description>
6. <link>http://www.example.com/main.html</link>
7. <lastBuildDate>Mon, 06 Sep 2010 00:01:00 +0000 </lastBuildDate>
8. <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
9. <ttl>1800</ttl>
10.
11. <item>
12. <title>Example entry</title>
13. <description> Here is some text containing an interesting description.
    </description>
14. <link>http://www.example.com/blog/post/1</link>
15. <guid isPermaLink="true">7bd204c6-1655-4c27-aeec-53f933c5395f</guid>
16. <pubDate>Sun, 06 Sep 2009 16:20:00 +0000</pubDate>
17. </item>
18.
19. </channel>
20. </rss>

```

Κώδικας 4.1.12: RSS σε XML μορφοποιημένο κείμενο

Πιο συγκεκριμένα, δημιουργείται μία λίστα από αντικείμενα `RSSItem`, τα οποία περιέχουν τις πιο σημαντικές πληροφορίες κάθε ανακοίνωσης, όπως αναφέρθηκε και παραπάνω (Κώδικας 4.1.13)^[30].

```

1. List<RSSItem> itemList = new ArrayList<RSSItem>();
2. Document doc = this.getDomElement(rss_url);
3. // Getting items array
4. NodeList items = doc.getElementsByTagName(TAG_ITEM);
5. // looping through each item
6. for(int i = 0; i < items.getLength(); i++){
7.     Element e1 = (Element) items.item(i);
8.     String title = this.getValue(e1, TAG_TITLE);
9.     String link = this.getValue(e1, TAG_LINK);
10.    String description = this.getValue(e1, TAG_DESCRIPTION);
11.    String pubdate = this.getValue(e1, TAG_PUB_DATE);
12.    RSSItem rssItem = new RSSItem(title, link, description, pubdate);
13.    // adding item to list
14.    itemList.add(rssItem);
15. }

```

Κώδικας 4.1.13: Δημιουργία λίστα με RSSItem αντικείμενα

Η λίστα αυτή τοποθετείται με τη βοήθεια ενός αντάπτορα σε ένα listview (κώδικας 4.1.14) και επίσης ενεργοποιείται η μέθοδος setOnClickListener. Έτσι, κατά την επιλογή μίας τυχαίας ανακοίνωσης δημιουργείται ένα νέο activity, στο οποίο αποστέλλονται ως παράμετροι ο υπερσύνδεσμος, ο τίτλος και η μέρα δημοσιοποίησης της ανακοίνωσης, καθώς και το όνομα και ο κωδικός χρήστη. Το activity αυτό παρουσιάζει την ανακοίνωση στην πλήρη της μορφή (Κώδικας 4.1.15).

```

1. ListAdapter adapter = new SimpleAdapter(ProptixiakiesAnakoinoiseis.this,
2.    rssItemList, R.layout.anakoinoiseis_list_item,
3.    new String[] {TAG_LINK, TAG_TITLE, TAG_PUB_DATE, TAG_DESCRIPTION},
4.    new int[] { R.id.page_url,R.id.title,R.id.pub_date,R.id.link });
5. // updating listview
6. setListAdapter(adapter);

```

Κώδικας 4.1.14: Ενημέρωση του ListView

```

1. listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
2.    public void onItemClick(AdapterView<?> parent, View view,
3.        int position, long id) {
4.        String str1 = ((TextView)view.findViewById(R.id.page_url)).getText()
5.            .toString();
6.        String str2 = ((TextView)view.findViewById(R.id.title)).getText().
7.            toString();
8.        String str3 = ((TextView)view.findViewById(R.id.pub_date)).getText()
9.            .toString();
10.       String str4 = ((TextView)view.findViewById(R.id.link)).getText().t
11.           oString();
12.       //Starting new intent
13.       Intent in = new Intent(ProptixiakiesAnakoinoiseis.this, Ana-
14.           kItemView.class);
15.       // passing data
16.       in.putExtra("url", str1);

```

```

12.         in.putExtra("title", str2);
13.         in.putExtra("date", str3);
14.         in.putExtra("desc", str4);
15.         in.putExtra("username", prefs.getString("username", null));
16.         in.putExtra("password", prefs.getString("password", null));
17.         startActivity(in);
18.     }
19. });

```

Κώδικας 4.1.15: Η μέθοδος `setOnClickListener`

GenikesAnakoinoseis.java

Η λειτουργικότητα της κλάσης αυτής είναι παρόμοια με την προηγούμενη με τη διαφορά ότι χειρίζεται τις γενικές ανακοινώσεις της σχολής. Αξίζει να σημειωθεί ότι η σύνδεση του χρήστη δεν κρίνεται απαραίτητη και έτσι ο έλεγχος συνδεσιμότητας δεν πραγματοποιείται.

AnakItemView.java

Αποτελεί το `activity`, στο οποίο παρουσιάζεται πλήρως το περιεχόμενο κάθε ανακοίνωσης είτε είναι γενική είτε είναι προπτυχιακών. Στην περίπτωση που η ανακοίνωση είναι γενική, χρησιμοποιείται το `Jsoup` για την σύνδεση στο `link` της ανακοίνωσης και την προσκόμιση του περιεχομένου της (κώδικας 4.1.16). Αντίθετα, αν πρόκειται για ανακοίνωση προπτυχιακών, απαιτείται επικοινωνία με τον εξυπηρέτη και μεταφορά των κωδικών του χρήστη για την προσκόμισή της.

```

1. Document doc = Jsoup.connect(params[0]).get();
2. Element element = doc.select("div[class=pf-content]").first();
3. String str = element.html();

```

Κώδικας 4.1.16: Προσκόμιση της περιγραφής της ανακοίνωσης με το `Jsoup`

Vathmoi.java

Η κλάση αυτή χειρίζεται τους βαθμούς του χρήστη. Απαραίτητη προϋπόθεση είναι η σύνδεσή του με τους προσωπικούς του κωδικούς. Για το λόγο αυτό κατά την εκκίνηση του συγκεκριμένου `activity` πραγματοποιείται έλεγχος συνδεσιμότητας, όπως περιγράφηκε και σε προηγούμενη κλάση. Η ανάλυση των δεδομένων πραγματοποιείται με τη βοήθεια της κλάσης `JSONArray`. Κάθε αντικείμενο του πίνακα περιέχει τρία πεδία, τον τίτλο, τον τύπο (υποχρεωτικό ή επιλογής) και το βαθμό του μαθήματος. Τα πεδία αυτά τοποθετούνται στα αντίστοιχα `TextViews` που αποτελούν στήλες ενός `TableLayout`. Υιοθετήθηκε αυτή η λογική, ώστε το αποτέλεσμα να μοιάζει με αυτό του `studentWeb` (Κώδικας 4.1.17).

```

1. JSONArray jsonArray = new JSONArray(str);
2. tv0.setText(jsonObject.getString("mathima"));
3. tv1.setText(jsonObject.getString("typos").substring(0, 1));
4. tv2.setText(jsonObject.getString("vathmos"));

```

Κώδικας 4.1.17: Ανάλυση δεδομένων στην κλάση **Vathmoi**

Programma.java

Και σε αυτό το activity απαιτείται σύνδεση για την προβολή των περιεχομένων του. Το activity έχει υλοποιηθεί με τον ViewPagerAdapter, όπου κάθε σελίδα περιέχει ένα διαφορετικό fragment και κάθε fragment αποτελείται από ένα TableLayout. Στην πραγματικότητα κάθε σελίδα αντιπροσωπεύει μία ημέρα της εβδομάδας, από Δευτέρα μέχρι και Παρασκευή. Με την επιλογή διαφορετικής σελίδας κάθε φορά ο Fragment Manager εναλλάσσει τα fragments στην οθόνη του τηλεφώνου. Κάθε fragment είναι υπεύθυνο για την αρχικοποίηση της δικιάς του προβολής (Κώδικας 4.1.18)^[31]. Κατά την εκκίνηση του activity επιλέγεται η τρέχουσα σελίδα ανάλογα με την τρέχουσα ημέρα της εβδομάδας. Στην περίπτωση του σαββατοκύριακου επιλέγεται η Δευτέρα ως τρέχουσα σελίδα (κώδικας 4.1.19).

```

1. view =inflater.inflate(R.layout.tab_triti,container,false);
2. prefs = this.getActivity().getSharedPreferences("login_options", Context.MODE_PRIVATE);
3. sch = prefs.getString("sch", null);
4. table = (TableLayout)view.findViewById(R.id.table_triti);
5. try {
6.     jsonArray = new JSONArray(sch);
7.     coursesDisplay(jsonArray.getJSONArray(1), table, 1, container, inflater);
8. } catch (JSONException e) {
9.     e.printStackTrace();
10. }

```

Κώδικας 4.1.18: Αρχικοποίηση του fragment για τη μέρα Τρίτη

```

1. Viewpager pager;
2. SlidingTabLayout tabs;
3. // Creating The ViewPagerAdapter and Passing Fragment Manager, Titles for the Tabs and Number Of Tabs.
4. ViewPagerAdapter adapter = new ViewPagerAdapter (getSupportFragmentManager(),Titles,Numboftabs);
5. // Assigning ViewPager View and setting the adapter
6. pager = (ViewPager) findViewById(R.id.pager);
7. pager.setAdapter(adapter);
8. Calendar calendar = Calendar.getInstance();
9. int day = calendar.get(Calendar.DAY_OF_WEEK);

```

```

10.
11. if((day==7) || (day == 1)){
12.     day = 2;
13. }
14. pager.setCurrentItem(day - 2);

```

Κώδικας 4.1.19: Αρχικοποίηση ViewPager

Η ανάλυση των δεδομένων πραγματοποιείται ξανά με τη χρήση της JSONArray κλάσης. Υπάρχουν 5 πίνακες, ένας για κάθε μέρα, με αντικείμενα που το καθένα αποτελείται από 5 πεδία, την ώρα διεξαγωγής του μαθήματος, το μάθημα, την αίθουσα που λαμβάνει χώρα το μάθημα, τους διδάσκοντες και τον τύπο του μαθήματος (διάλεξη ή εργαστήριο). Κάθε πεδίο τοποθετείται στο αντίστοιχο TextView στην αντίστοιχη στήλη στον πίνακα. Κάθε TextView, εκτός του τύπου του μαθήματος, υλοποιεί μια ξεχωριστή ενέργεια κατά το πάτημά του. Αυτό επιτυγχάνεται με τη βοήθεια των κλάσεων SpannableString και ClickableSpan (Κώδικας 4.1.20). Συγκεκριμένα:

- Κατά το πάτημα πάνω στην ώρα του μαθήματος εκκινείται το ημερολόγιο του τηλεφώνου και προτρέπεται η δημιουργία ενός συμβάντος σε αυτό με τις εξής προεπιλογές (Κώδικας 4.1.21):
 - Την ημέρα διεξαγωγής του μαθήματος, ανάλογα με τη σελίδα που είναι ανοικτή.
 - Την ώρα έναρξης και την ώρα λήξης του μαθήματος με τη χρήση της κλάσης SimpleDateFormat.
 - Εβδομαδιαία συχνότητα και συνολική διάρκεια 12 εβδομάδες (όσες και οι επίσημες μέρες διεξαγωγής μαθημάτων ανά εξάμηνο).
 - Την τοποθεσία διεξαγωγής του μαθήματος.
 - Τον τίτλο του μαθήματος.
 - Τον τύπο του μαθήματος.
- Κατά το πάτημα πάνω στον τίτλο του μαθήματος, πραγματοποιείται ανακατεύθυνση στο activity MathimataProfil, το οποίο παρουσιάζει το προφίλ του κάθε μαθήματος (Κώδικας 4.1.22).
- Κατά το πάτημα πάνω στην αίθουσα διδασκαλίας, πραγματοποιείται ανακατεύθυνση στο activity Xartis, στο οποίο φαίνεται η τοποθεσία της αίθουσας πάνω στο χάρτη σημειωμένη με ειδική πινέζα (Κώδικας 4.1.23).
- Κατά το πάτημα πάνω στον διδάσκοντα του μαθήματος, πραγματοποιείται ανακατεύθυνση στο activity Profil, το οποίο παρουσιάζει το προφίλ του (Κώδικας 4.1.24).

```

1. SpannableString ss0 = new SpannableString(ora);
2. ClickableSpan clickableSpan0 = new ClickableSpan() {
3.     @Override
4.     public void onClick(View textView) {
5.         Intent intent = new Intent(Intent.ACTION_EDIT);
6.         startActivity(intent);

```

```

7.     }
8.   };
9.   ss0.setSpan(clickableSpan0, 0, ss0.length(), Spanned.SPAN_EXCLUSIVE_EXCLUS
IVE);
10.  tv0.setText(ss0);
11.  tv0.setMovementMethod(LinkMovementMethod.getInstance());

```

Κώδικας 4.1.20: Αρχικοποίηση ενέργειας σε κείμενο

```

1.  Intent intent = new Intent(Intent.ACTION_EDIT);
2.  intent.setType("vnd.android.cursor.item/event");
3.  intent.putExtra("beginTime", time1.getTime());
4.  intent.putExtra("allDay", false);
5.  intent.putExtra("rrule", "FREQ=WEEKLY;COUNT=12;WKST=SU;BYDAY="+ date);
6.  intent.putExtra("endTime", time2.getTime());
7.  intent.putExtra(CalendarContract.Events.EVENT_LOCATION, aith-
ousa.replaceAll("\n", " "));
8.  intent.putExtra(CalendarContract.Events.TITLE, e11.text());
9.  intent.putExtra(CalendarContract.Events.DESCRPTION, typos);
10. startActivity(intent);

```

Κώδικας 4.1.21: Δημιουργία συμβάντος στο ημερολόγιο της συσκευής

```

1.  Intent i = new Intent(getActivity(), MathimataProfil.class);
2.  i.putExtra("lesson_name", e11.select("a[href]").get(0).attr("abs:href"));
3.  startActivity(i);

```

Κώδικας 4.1.22: Ανακατεύθυνση στο προφίλ του μαθήματος

```

1.  final String str = tv3.getText().toString().replaceAll("\n", " ");
2.  Intent i = new Intent(getActivity(), Xartis.class);
3.  i.putExtra("find", "true");
4.  i.putExtra("aith", str);
5.  startActivity(i);

```

Κώδικας 4.1.23: Ανακατεύθυνση στους χάρτες

```

1.  Intent in = new Intent(getActivity(), Profil.class);
2.  in.putExtra("url", prof);
3.  startActivity(in);

```

Κώδικας 4.1.24: Ανακατεύθυνση στο προφίλ του διδάσκοντα

MyProgramma.java

Η λειτουργικότητα της κλάσης MyProgramma είναι η ίδια με την προηγούμενη με τη διαφορά ότι τώρα παρουσιάζεται το προσωπικό πρόγραμμα του χρήστη σύμφωνα

με τη δήλωση μαθημάτων που έχει κάνει κατά το τρέχον εξάμηνο. Σε κάθε σελίδα εμφανίζονται μόνο τα μαθήματα που έχει δηλώσει. Κατά την πρώτη είσοδο του χρήστη στο activity και εφόσον είναι ήδη συνδεδεμένος πραγματοποιείται επικοινωνία με τον εξυπηρέτη, όπου προσκομίζεται η δήλωση μαθημάτων του ως ένας JSON πίνακας. Στον πίνακα περιέχονται αντικείμενα (μαθήματα) με το πεδίο “mathima”, το οποίο περιλαμβάνει τον κωδικό κάθε μαθήματος. Σύμφωνα με τους κωδικούς και την βοήθεια των προπτυχιακών μαθημάτων (μέθοδος findCourses), δημιουργείται ένα ArrayList, όπου αποθηκεύονται τα μαθήματα της δήλωσης του χρήστη. Στη συνέχεια η δήλωση αποθηκεύεται στα sharedPreferences, ώστε σε κάθε επόμενη είσοδό του στο activity να μην επαναλαμβάνεται αυτή η διαδικασία (scheduled = true, κώδικας 4.1.25).

```
1. editor = prefs1.edit();
2. editor.clear();
3. editor.putString("scheduled", "true");
4. editor.putString("username", prefs.getString("username", null));
5. editor.putString("courses", maths);
6. editor.commit();
```

Κώδικας 4.1.25: Αποθήκευση της δήλωσης μαθημάτων στη συσκευή

Τέλος, τα μαθήματα αντιστοιχίζονται στις αντίστοιχες προβολές όπως και στην κλάση Programma. Η μόνη διαφορά είναι ότι κάθε μάθημα ελέγχεται, εάν ανήκει στη δήλωση μαθημάτων του χρήστη, προτού τοποθετηθεί (κώδικας 4.1.26).

```
1. if(e11.text().toLowerCase().equals(list.get(1).toLowerCase())) {...}
```

Κώδικας 4.1.26: Έλεγχος για το αν ανήκει στη λίστα μαθημάτων της δήλωσης

Xartis.java

Για την ενσωμάτωση των χαρτών της Google στην εφαρμογή είναι απαραίτητη η δημιουργία ενός Google Map Android V2 κλειδιού^{[32] [33]}. Για την επίτευξη αυτού πρέπει να υλοποιηθούν τα επόμενα βήματα:

- Στην διεύθυνση όπου βρίσκεται η jdk έκδοση, πληκτρολογείται η εντολή του κώδικα 4.1.27. Από την έξοδο της εντολής αντιγράφεται το SHA1 κλειδί, το οποίο χρησιμεύει στη δημιουργία του Google Map Android V2 κλειδιού.

```
1. keytool -list -v -keystore ~/.android/debug.keystore -
   alias androiddebugkey -storepass android -keypass android
```

Κώδικας 4.1.27

- Στη συνέχεια πραγματοποιείται σύνδεση στο Google API's Console με έναν Google λογαριασμό και δημιουργία ενός καινούργιου project. Από το τμήμα "APIs" ενεργοποιείται το "Google Map Android API V2" και μετά από το τμήμα "Credentials" επιλέγεται το κουμπί "Create new Key". Εδώ πρέπει να τοποθετηθεί το SHA1 κλειδί και με το πάτημα του κουμπιού "Create" γίνεται λήψη του Google Map Android V2 κλειδιού.
- Τέλος, πρέπει
 - το κλειδί να δηλωθεί στο AndroidManifest.xml της εφαρμογής (Κώδικας 4.1.28) και
 - να προστεθούν κάποιες άδειες (permissions), οι οποίες είναι απαραίτητες για τη χρήση των χαρτών, στο AndroidManifest.xml της εφαρμογής (Κώδικας 4.1.29).

```

1. <meta-data
2.   android:name="com.google.android.maps.v2.API_KEY"
3.   android:value="Your Google Maps API V2 Key" />

```

Κώδικας 4.1.28: Δήλωση του Google Maps API V2 κλειδιού στο AndroidManifest

```

1. <uses-permission android:name="android.permission.INTERNET" />
2. <uses-permission android:name="android.permission.
   WRITE_EXTERNAL_STORAGE"/>
3. <uses-permission android:name="com.google.android.providers.
   gsf.permission.READ_GSERVICES"/>
4. <uses-permission android:name="android.permission.
   ACCESS_COARSE_LOCATION" />
5. <uses-permission android:name="android.permission.
   ACCESS_FINE_LOCATION" />

```

Κώδικας 4.1.29: Permissions στο AndroidManifest για τη χρήση των χαρτών

Για την υλοποίηση της κλάσης Xartis χρησιμοποιήθηκαν δύο προβολές, ένα fragment, όπου αναπαριστάται ο χάρτης, και ένα ListView, όπου αναπαρίστανται χώροι του πανεπιστημίου, κ.α. Το ListView περιέχεται μέσα σε ένα DrawerLayout. Η ιδιαιτερότητα του DrawerLayout είναι ότι παραμένει αόρατο στην κύρια οθόνη της συσκευής, μέχρι να πατηθεί το αντίστοιχο κουμπί (Κώδικας 4.1.30). Κατά την εκκίνηση του activity πραγματοποιείται επικοινωνία με τον εξυπηρέτη, ώστε να προσκομιστούν οι συντεταγμένες των σημείων. Στην συνέχεια τοποθετούνται στο ListView με τη βοήθεια της κλάσης NavDrawerListAdapter, η οποία δημιουργεί έναν αντάπτορα. Πιο συγκεκριμένα, οι συντεταγμένες των σημείων επιστρέφονται ως ένα αλφαριθμητικό και τοποθετούνται σε ένα ArrayList με NavDrawerItem αντικείμενα. Το item[0] περιέχει το όνομα του σημείου και το item[1] τις συντεταγμένες του (Κώδικας 4.1.31). Στο χάρτη αναπαριστάται κάθε φορά ένα σημείο. Όταν επιλέγεται ένα σημείο από το ListView, απομακρύνεται το προηγούμενο σημείο και εμφανίζεται στην οθόνη το σημείο που επιλέχθηκε (Κώδικας 4.1.32).


```

1. boolean drawerOpen = mDrawerLayout.isDrawerOpen(mDrawerList);
2. if (drawerOpen) {
3.     mDrawerLayout.closeDrawer(mDrawerList);
4. } else { mDrawerLayout.openDrawer(mDrawerList); }

```

Κώδικας 4.1.30: Άνοιγμα και κλείσιμο του listview

```

1. final ArrayList<NavDrawerItem> itemList = new Ar-
   rayList<NavDrawerItem>();
2. String[] simeia = str.split("\n");
3. for(int i=0; i < simeia.length; i++){
4.     String[] item = simeia[i].split(":");
5.     NavDrawerItem navDrawerItem = new NavDrawerItem(item[0], item[1]);
6.     itemList.add(navDrawerItem);
7. }

```

Κώδικας 4.1.31: Ανάλυση των συντεταγμένων σημείων

```

1. myMarker.remove();
2. final String[] dirs = coord.split(",");
3. final Double lat = Double.parseDouble(dirs[0]);
4. final Double lng = Double.parseDouble(dirs[1]);
5. myMarker = googleMap.addMarker(new MarkerOptions()
6.     .position(new LatLng(lat, lng))
7.     .title(title));
8. myMarker.setIcon(BitmapDescriptorFactory.fromResource
   (R.mipmap.red_pin));
9. CameraUpdate cu = CameraUpdateFactory.newLatLngZoom
   (new LatLng(lat, lng), 13);
10. googleMap.animateCamera(cu);
11. myMarker.showInfoWindow();

```

Κώδικας 4.1.32: Αναπαράσταση σημείου πάνω στο χάρτη

Didaskontes.java

Το activity *Didaskontes* παρουσιάζει τους διδάσκοντες του τμήματος με τη βοήθεια ενός *ExpandableListView*. Η χρήση του *ExpandableListView* κρίθηκε απαραίτητη, ώστε οι κατηγορίες διδασκόντων που υπάρχουν στο τμήμα (καθηγητές, αναπληρωτές καθηγητές, κ.α.) να ομαδοποιηθούν κατάλληλα. Η κλάση *DidaskontesExpandableListAdapter* περιέχει τον constructor για τον αντίστοιχο του *ExpandableListView*. Το περιεχόμενο της ιστοσελίδας των διδασκόντων επιστρέφεται από τον εξυπηρέτη ως ένας JSON πίνακα, ο οποίος περιέχει διάφορα αντικείμενα. Κάθε αντικείμενο περιέχει 3 πεδία, το όνομα του διδάσκοντα, την κατηγορία που ανήκει, καθώς και τον υπερσύνδεσμο για το προφίλ του στον ιστότοπο της σχολής. Η ανάλυση τους γίνεται μέσα στη μέθοδο *getDidaskontes(String str)*, η οποία αρχικοποιεί δύο λίστες:

- ο την `listDataHeader`, η οποία είναι μια `ArrayList` και περιέχει τα ονόματα όλων των κατηγοριών.
- ο την `listDataChild`, η οποία είναι μια λίστα `HashMap` και περιέχει ως κλειδιά τα ονόματα των κατηγοριών και ως τιμή σε κάθε κλειδί μία `ArrayList` με τα ονόματα και τους υπερσυνδέσμους των διδασκόντων-παιδιών υλοποιημένα με `HashMap`.

Τέλος, εφαρμόστηκε η μέθοδος `setOnChildClickListener`, με την οποία πατώντας πάνω σε κάποιον διδάσκοντα δημιουργείται ένα καινούριο `activity` που παρουσιάζει το προφίλ του. Η `get(groupPosition)` και η `get(childPosition)` επιστρέφουν την ακριβή θέση του στοιχείου μέσα στην προβολή. Ο έλεγχος για `null` ή εάν είναι άδαιο το αλφαριθμητικό πραγματοποιείται για την περίπτωση που ο διδάσκων δεν διαθέτει κάποιον υπερσύνδεσμο για το προφίλ του (κώδικας 4.1.34).

```
1. ArrayList listDataHeader = new ArrayList<String>();
2. HashMap listDataChild = new HashMap<String,ArrayList<HashMap<String, String>>>();
```

Κώδικας 4.1.33: Οι δυο λίστες της μεθόδου `getDidaskontes`

```
1. expListView.setOnChildClickListener
   (new ExpandableListView.OnChildClickListener() {
2.
3.     @Override
4.     public boolean onChildClick(ExpandableListView parent, View v,
5.         int groupPosition, int childPosition, long id) {
6.         String str = listDataChild.get(listDataHeader.get(groupPosition))
7.             .get(childPosition).get("PROFIL");
8.         if(str !=null && !str.isEmpty()) {
9.             Intent in = new Intent(Didaskontes.this, Profil.class);
10.            in.putExtra("url", str);
11.            startActivity(in);
12.        }
13.        else{
14.            Toast.makeText(Didaskontes.this, "Δυστυχώς το προφίλ του διδά-
15.            σκοντα δεν βρέθηκε", Toast.LENGTH_LONG).show();
16.        }
17.        return false;
18.    });
```

Κώδικας 4.1.34: Υλοποίηση μεθόδου `setOnChildClickListener`

Profil.java

Η κλάση αυτή παρουσιάζει το προφίλ κάθε διδάσκοντα. Κατά την εκκίνηση της λαμβάνεται η παράμετρος “URL” που περιέχει τον υπερσύνδεσμο της διεύθυνσης, όπου βρίσκεται το προφίλ του διδάσκοντα στον ιστότοπο της σχολής.

```
1. Intent i = getIntent();
2. profileURL = i.getStringExtra("url");
```

Κώδικας 4.1.35: Λήψη url διδάσκοντα

Στη συνέχεια με τη χρήση του Jsoup γίνεται λήψη των περιεχόμενων της σελίδας και ανάλυσή τους στα αντίστοιχα TextViews. Για την λήψη της εικόνας του διδάσκοντα χρησιμοποιείται η βιβλιοθήκη Volley. Όπως φαίνεται και στον κώδικα 4.1.36, απαιτούνται μόνο δύο γραμμές για την εφαρμογή της μεθόδου.

```
1. ImageLoader imageLoader = ApplicationController.getInstance().getImageLoader();
2. // NetworkImageView
3. imageView.setImageUrl(imgSrc, imageLoader);
```

Κώδικας 4.1.36: Χρήση Volley για την λήψη της εικόνας του διδάσκοντα

Mathimata.java

Το activity Mathimata ακολουθεί την λογική του activity Didaskontes. Εδώ παρουσιάζονται όλα τα προπτυχιακά μαθήματα της σχολής με βάση το γνωστικό αντικείμενο, στο οποίο ανήκουν. Το περιεχόμενο της αντίστοιχης σελίδας στον ιστότοπο της σχολής επιστρέφεται από τον εξυπηρέτη σε ένα JSON πίνακα, όπου περιέχονται όλα τα μαθήματα καθώς και οι κατηγορίες μαθημάτων. Στην μέθοδο getMathimata(String str) πραγματοποιείται η αρχικοποίηση της λίστας με τις κατηγορίες μαθημάτων και των λιστών-παιδιών τους. Η κλάση MathimataExpandableListAdapter περιέχει τον constructor του αντάπτορα. Τέλος, κατά το πάτημα ενός μαθήματος δημιουργείται ένα καινούργιο activity, στο οποίο μεταβιβάζεται ως παράμετρος το όνομα του μαθήματος.

MathimataProfil.java

Το activity αυτό υλοποιήθηκε με τη χρήση ViewPagerAdapter, όπου υπάρχουν 2 σελίδες. Μία σελίδα παρουσιάζει τις πληροφορίες για το μάθημα (Γνωστικό αντικείμενο, Εξάμηνο, Υπεύθυνοι μαθήματος, κ.α.) και μία άλλη σελίδα παρουσιάζει το ατομικό πρόγραμμα του μαθήματος. Και εδώ υιοθετήθηκαν λογικές που περιγράφηκαν παραπάνω.

Lesxi.java

Σε αυτή τη κλάση τα περιεχόμενα φορτώνονται από τα sharedPreferences και η ανάλυσή τους γίνεται με τη κλάση JSONObject (Κώδικας 4.1.37). Μαζί με το TextView με τις πληροφορίες για τη σίτιση περιλαμβάνονται και δύο TextViews με SpannableStrings, τα οποία περιέχουν τους υπερσυνδέσμους για τα μενού σίτισης της

τρέχουσας και της επόμενης εβδομάδας. Κατά το πάτημα πάνω τους δημιουργείται η κλάση MenuLesxis (Κώδικας 4.1.38).

```
1. JSONObject jsonobject = new JSONObject(params[0]);
2. String str = jsonobject.getString("keimeno") + "Παρακάτω μπορείτε να δείτε
   το πρόγραμμα σίτισης.\n";
3. String link_str1 = jsonobject.getString("link1");
4. String link_str2 = jsonobject.getString("link2");
```

Κώδικας 4.1.37: Ανάλυση δεδομένων στη κλάση Lesxi

```
1. SpannableString ss1 = new SpannableString("Καινούργιο πρόγραμμα
   σίτισης");
2. ClickableSpan clickableSpan1 = new ClickableSpan() {
3.     @Override
4.     public void onClick(View textView) {
5.         Intent i = new Intent (Lesxi.this, MenuLesxis.class);
6.         i.putExtra("URL", link_str2);
7.         startActivity(i);
8.     }
9. };
10. ss1.setSpan(clickableSpan1, 0, ss1.length()-
    1, Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
11. TextView textView2 = (TextView) findViewById(R.id.textView6);
12. textView2.setText(ss1);
13. textView2.setMovementMethod(LinkMovementMethod.getInstance());
```

Κώδικας 4.1.38: Δημιουργία SpannableString

MenuLexis.java

Κατά την εκκίνηση του συγκεκριμένου activity γίνεται λήψη της παραμέτρου “URL” που περιέχει τον υπερσύνδεσμο, στον οποίο βρίσκεται το πρόγραμμα σίτισης που ζητήθηκε. Το πρόγραμμα σίτισης παρέχεται στη μορφή εγγράφου doc. Έτσι, με τη μέθοδο showDoc(String str) πραγματοποιείται φόρτωση του εγγράφου με τη βοήθεια των εγγράφων της Google και παρουσίασής του σε ένα WebView.

```
1. Intent i = getIntent();
2. String str = i.getStringExtra("URL");
3.
4. private void showDoc(String str){
5.     WebView webview = new WebView(this);
6.     setContentView(webview);
7.     webview.getSettings().setJavaScriptEnabled(true);
8.     webview.loadUrl("http://docs.google.com/gview?embedded=true&url="+str);
9. }
```

Κώδικας 4.1.39: Κώδικας MenuLexis.java

Leoforeia.java

Το activity *Leoforeia* περιλαμβάνει όλα τα δρομολόγια των αστικών λεωφορείων του δήμου Βόλου. Τα περιεχόμενα φορτώνονται από τα *sharedpreferences* και αναλύονται με τη βοήθεια της κλάσης *JSONArray*. Κάθε αντικείμενο του πίνακα περιλαμβάνει 3 πεδία, το νούμερο της διαδρομής, το αρχικό δρομολόγιο και το δρομολόγιο επιστροφής. Τα περιεχόμενα αποθηκεύονται σε μία λίστα από *BusItem* αντικείμενα και στην συνέχεια προσαρμόζονται στο *ListView* με την βοήθεια του αντάπτορα *BusListAdapter*. Αξίζει να σημειωθεί ότι μέσα στη μέθοδο *getView* της *BusListAdapter* έχουν υλοποιηθεί δύο *setOnClickListener* μέθοδοι, μία για κάθε δρομολόγιο της γραμμής. Κάτι τέτοιο κρίθηκε απαραίτητο να υλοποιηθεί μέσα στη μέθοδο *getView*, η οποία είναι υπεύθυνη για τη δημιουργία της προβολής της λίστας. Το *holder.textLine1* κρατάει τη θέση του αντίστοιχου *TextView*. Κατά το πάτημά του δημιουργείται ένα καινούργιο activity, στο οποίο αποστέλλονται το όνομα του δρομολογίου, καθώς και η θέση του μέσα στην αντίστοιχη γραμμή της λίστας (0 αν είναι το αρχικό δρομολόγιο και 1 αν είναι το δρομολόγιο επιστροφής (Κώδικας 4.1.40).

```
1. holder.textLine1.setOnClickListener(new View.OnClickListener() {
2.
3.     @Override
4.     public void onClick(View v) {
5.         String line = BusListAdapter.listMap.get(holder.position).get("dromologio_1");
6.         Intent i = new Intent(context, Dromologio.class);
7.         i.putExtra("URL", line);
8.         i.putExtra("POSITION", String.valueOf(0));
9.         context.startActivity(i);
10.    }
11. });
```

Κώδικας 4.1.40: Υλοποίηση *setOnClickListener* μέσα στη μέθοδο *getView*

Dromologio.java

Στη συγκεκριμένη κλάση γίνεται η παρουσίαση του δρομολογίου που ζητήθηκε στο activity *Leoforeia*. Κατά την εκκίνησή της πραγματοποιείται η λήψη των παραμέτρων "URL, POSITION" που αντιστοιχούν στο όνομα του δρομολογίου και στη θέση του στην αντίστοιχη γραμμή της λίστας. Επίσης, δημιουργείται επικοινωνία με τον εξυπηρέτη, ώστε να επιστραφεί ο υπερσύνδεσμος με τη διεύθυνση του δρομολογίου. Στη συνέχεια με τη βοήθεια ενός *WebView* και των εγγράφων της *Google*, το δρομολόγιο εμφανίζεται στη οθόνη, όπως ακριβώς συμβαίνει και στην κλάση *MenuLesxis*. Η μόνη διαφορά είναι ότι τα δρομολόγια των λεωφορείων είναι σε μορφή pdf, κάτι το οποίο δεν επηρεάζει καθόλου την υλοποίηση.

PeriStep.java & About.java & Fisiognomia.java

Τέλος, οι τρεις αυτές κλάσεις παρουσιάζουν πληροφορίες για το τμήμα και για την ίδια την εφαρμογή. Η κλάση PeriStep αποτελεί το ενδιάμεσο βήμα για την επιλογή των άλλων δύο. Στην κλάση About παρέχονται γενικές πληροφορίες για την εφαρμογή, ενώ στην κλάση Fisiognomia παρουσιάζεται η φυσιognωμία του τμήματος, όπου με τη χρήση του Jsoup λαμβάνονται τα περιεχόμενα της αντίστοιχης σελίδας από τον ιστότοπο της σχολής.

Λογική σχεδιασμού

Threads

Η χρήση νημάτων στην υλοποίηση της εφαρμογής κρίθηκε αναγκαία για λόγους εξοικονόμησης σε χρόνο. Με τον τρόπο αυτό ο χρόνος αναμονής δεν ξεπερνά το χρόνο απόκρισης του πιο χρονοβόρου νήματος (χρόνος για την προσκόμιση των περιεχομένων της πιο χρονοβόρας σελίδας). Σε αντίθετη περίπτωση, ο χρόνος αναμονής θα ήταν ανάλογος των διαδικασιών που εκτελούνταν. Αυτό θα είχε σαν αποτέλεσμα την αρνητική εμπειρία του χρήστη.

Συγκεκριμένα, ο συνολικός χρόνος αναμονής κατά την εκκίνηση της εφαρμογής χωρίς τη χρήση νημάτων θα ξεπερνούσε τα 9 δευτερόλεπτα, εάν λάβουμε υπόψη τους χρόνους προσπέλασης της εικόνας 4.1.2, καθώς και το χρόνο επεξεργασίας των δεδομένων στον εξυπηρέτη. Αντίθετα, με τη χρήση νημάτων ο μέσος χρόνος αναμονής είναι 3.5 δευτερόλεπτα.

programma-sitisis	200	document	Other	20.7 KB	206 ms
?page_id=54	200	document	Other	82.8 KB	3.09 s
?cat=24	200	document	Other	47.6 KB	1.60 s
index.php?optio...	200	document	Other	4.6 KB	209 ms
?page_id=7758	200	document	Other	118 KB	3.14 s

Εικόνα 4.1.2: Μέτρηση χρόνων προσπέλασης ιστοσελίδων με τον περιηγητή Google Chrome

Στην περίπτωση της σύνδεσης στην εφαρμογή χρησιμοποιούνται δύο νήματα. Το πιο χρονοβόρο από τα δύο είναι υπεύθυνο για την προσκόμιση των προπτυχιακών ανακοινώσεων και του εβδομαδιαίου προγράμματος από το site της σχολής. Ο χρόνος που εξοικονομείται από την εφαρμογή δύο νημάτων, είναι ο χρόνος που διαρκεί η εργασία του δεύτερου νήματος. Όπως φαίνεται και από την εικόνα 4.1.3, ο χρόνος αυτός υπολογίζεται περίπου στο 1 δευτερόλεπτο με βάση το χρόνο προσπέλασης των δύο σελίδων αλλά και το χρόνο επεξεργασίας των δεδομένων στον εξυπηρέτη. Συνολικά η διαδικασία σύνδεσης διαρκεί περίπου 6 δευτερόλεπτα.

login.asp	302	text/html	Other	321 B	366 ms
stud_CResults.as...	200	document	/unist...	64.1 KB	391 ms

Εικόνα 4.1.3: Μέτρηση χρόνων προσπέλασης ιστοσελίδων με τον περιηγητή Google Chrome

Σε κάθε περίπτωση, ο χρόνος προσκόμισης των περιεχόμενων των σελίδων είναι άμεσα εξαρτημένος από το χρόνο προσπέλασης των σελίδων του ιστότοπου της σχολής.

Αποθήκευση περιεχομένων

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, τα περισσότερα δεδομένα που επιστρέφονται από τον εξυπηρέτη αποθηκεύονται ως αλφαριθμητικά στα sharedpreferences της εφαρμογής. Με τον τρόπο αυτό βελτιώνεται ο χρόνος περιήγησης μέσα στην εφαρμογή. Διαφορετικά, για κάθε activity η επικοινωνία με τον εξυπηρέτη θα ήταν αναπόφευκτη, ώστε να προσκομιστούν τα δεδομένα που χρειάζονται και στην συνέχεια να γίνει αντιστοίχσή τους στη διεπαφή χρήστη. Ο χρόνος που εξοικονομείται σε αυτήν την περίπτωση είναι ο χρόνος επικοινωνίας με τον εξυπηρέτη, όπου σε κάποιες περιπτώσεις μπορεί να φτάσει μέχρι και 5 δευτερόλεπτα.

4.2 Εξυπηρέτης

Πριν την περιγραφή του κώδικα του εξυπηρέτη που υλοποιήθηκε, κρίνεται σκόπιμο να αναφερθούν ορισμένες έννοιες που θα χρησιμοποιηθούν στη συνέχεια.

- **Curl**^[34]: Είναι ένα project, το οποίο παρέχει βιβλιοθήκη που επιτρέπει τη σύνδεση και την επικοινωνία με εξυπηρέτες υποστηρίζοντας πληθώρα πρωτοκόλλων. Η libcurl αυτή τη στιγμή υποστηρίζει http, https, ftp, ldap, κ.α. πρωτόκολλα.
- **DOMDocument**^[35]: Είναι η κλάση που παρουσιάζει ένα ολοκληρωμένο HTML ή XML έγγραφο.
- **Self-Signed certificate**^[36]: Στην κρυπτογραφία ένα self-signed πιστοποιητικό είναι ένα πιστοποιητικό που έχει υπογραφεί από τον ίδιο τον φορέα, του οποίου την ταυτότητα πιστοποιεί.
- **RSA**^[37]: Είναι ένας κρυπταλγόριθμος ασύμμετρου κλειδιού, το όνομα του οποίου προέρχεται από τους δημιουργούς του, Ron Rivest, Adi Shamir and Len Adleman.
- **Triple-DES**^[38]: Είναι ένα κρυπτοσύστημα, το οποίο αποτελείται από τρία συστήματα DES σε σειρά. Η έξοδος δηλαδή του πρώτου συστήματος είναι η είσοδος του δεύτερου, του οποίου η έξοδος είναι η είσοδος του τρίτου.
- **DES (Data Encryption Standard)**^[39]: Είναι ο κρυπταλγόριθμος που έχει επιλεγεί ως επίσημο Ομοσπονδιακό Πρότυπο Επεξεργασίας Πληροφοριών διεθνώς.

Γενική περιγραφή

Υλοποίηση SSL στο λογισμικό XAMPP

Για την υλοποίηση του πρωτοκόλλου SSL στον εξυπηρέτη απαιτούνται τα παρακάτω βήματα^[40]^[41]:

- **Δημιουργία RSA κλειδιού.** Πρόκειται για ένα RSA κλειδί 1024 bit, το οποίο είναι κρυπτογραφημένο με βάση το σύστημα Triple-DES και αποθηκεύεται στη μορφή PEM, ώστε να μπορεί να διαβαστεί ως ASCII κείμενο.

```
1. openssl genrsa -des3 -out server.key 1024
```

Κώδικας 4.2.1: Εντολή για τη δημιουργία του RSA κλειδιού

- **Παραγωγή ενός CSR (Certificate Signing Request).** Κατά την παραγωγή του CSR, απαιτούνται ορισμένες πληροφορίες, όπως το όνομα, η χώρα, όνομα εταιρίας, κ.α. Αυτές οι πληροφορίες αποτελούν X.509 χαρακτηριστικά του πιστοποιητικού.

```
1. openssl req -new -key server.key -out server.csr
```

Κώδικας 4.2.2: Εντολή για τη δημιουργία του CSR

- **Απενεργοποίηση του passphrase από το κλειδί.** Σε αντίθετη περίπτωση ο Apache εξυπηρέτης θα απαιτεί το passphrase κάθε φορά που εκκινείται.

```
1. cp server.key server.key.org  
2. openssl rsa -in server.key.org -out server.key
```

Κώδικας 4.2.3: Εντολές για τη απενεργοποίηση passphrase

- **Παραγωγή Self-Signed πιστοποιητικού.** Ο SSL εξυπηρέτης που θα υλοποιηθεί θα χρησιμοποιεί αυτό το πιστοποιητικό. Την παραγωγή ενός πιστοποιητικού για 365 μέρες υλοποιεί η επόμενη εντολή.

```
1. openssl x509 -req -days 365 -in server.csr -signkey server.key -  
out server.crt
```

Κώδικας 4.2.4: Εντολή για την παραγωγή Self-Signed πιστοποιητικού

— **Εγκατάσταση του προσωπικού κλειδιού και πιστοποιητικού στο XAMPP.**

```
1. cp server.crt /opt/lampp/etc/ssl.crt/server.crt
2. cp server.key /opt/lampp/etc/ssl.key/server.key
```

Κώδικας 4.2.5: Εντολές για την εγκατάσταση του προσωπικού κλειδιού και πιστοποιητικού στο XAMPP

— **Έλεγχος για την ενεργοποίηση του mod_ssl στο αρχείο /opt/lampp/etc/httpd.conf.**

```
1. LoadModule ssl_module modules/mod_ssl.so
```

Κώδικας 4.2.6: Έλεγχος ύπαρξης της εντολής στο httpd.conf

— **Προσθήκη virtual host στο αρχείο /opt/lampp/etc/extras/httpd-ssl.conf**

```
1. <virtualhost default:443>
2.     ServerName localhost.myserver.com
3.     ServerAlias localhost.myserver.com *.localhost.myserver.com
4.     ServerAdmin admin@localhost
5.
6.     DocumentRoot "/opt/lampp/htdocs/server/"
7.     DirectoryIndex index.php
8.
9.     ErrorLog /opt/lampp/logs/openenprs.local.error.log
10.    CustomLog /opt/lampp/logs/openenprs.local.access.log combined
11.
12.    SSLEngine on
13.    SSLCertificateFile /opt/lampp/etc/ssl.crt/openenprs.crt
14.    SSLCertificateKeyFile /opt/lampp/etc/ssl.key/openenprs.key
15.
16.    BrowserMatch ".*MSIE.*" nokeepalive ssl-unclean-shutdown
17.                downgrade-1.0 force-response-1.0
18. </virtualhost>
```

Κώδικας 4.2.7: Προσθήκη virtual host στο httpd-ssl.conf

Υλοποίηση εξυπηρέτη

Ο εξυπηρέτης υλοποιήθηκε στη γλώσσα προγραμματισμού php. Σκοπός του είναι να δέχεται κάποιες ετικέτες (tags) από την εφαρμογή και αναλόγως την ετικέτα να επιστρέφει στην εφαρμογή κάποια δεδομένα. Επίσης φιλοξενεί και δύο βάσεις δεδομένων, οι οποίες περιέχουν τα pages ids των προπτυχιακών μαθημάτων και τις συντεταγμένες διάφορων σημείων αντίστοιχα.

Σύνδεση στον ιστότοπο της σχολής και το studentWeb

Το όνομα και ο κωδικός χρήστη αποστέλλονται ως παράμετροι με τα ονόματα username και password αντίστοιχα (Κώδικας 4.2.8).

```
1. $username = $_POST['username'];  
2. $password = $_POST['password'];
```

Κώδικας 4.2.8: Λήψη του ονόματος και του κωδικού χρήστη

Ο κώδικας 4.2.9 υλοποιεί την σύνδεση και στην συνέχεια την προσπέλαση οποιασδήποτε σελίδας απαιτεί σύνδεση στον ιστότοπο της σχολής. Όπως και στον ιστότοπο της σχολής, για την σύνδεση είναι απαραίτητη η δημιουργία cookies, τα οποία διαγράφονται στο τέλος της σύνδεσης. Για την αποστολή των κωδικών χρήστη και των λοιπών παραμέτρων που απαιτούνται για την σύνδεση χρησιμοποιείται η μέθοδος POST. Εδώ αξίζει να τονιστεί ότι αφαιρέθηκε το redirect στην αρχική σελίδα του ιστότοπου από τις παραμέτρους (POSTFIELDS) για λόγους εξοικονόμησης χρόνου. Τέλος, τα περιεχόμενα της σελίδας επιστρέφονται ως αλφαριθμητικό (string) στη μεταβλητή \$resp, θέτοντας την επιλογή CURLOPT_RETURNTRANSFER με 1.

```
1. $ch = curl_init();  
2. curl_setopt($ch, CURLOPT_URL,$login_url);  
3. curl_setopt($ch, CURLOPT_USERAGENT, $agent);  
4. curl_setopt($ch, CURLOPT_POST, 1);  
5. curl_setopt($ch, CURLOPT_POSTFIELDS,$POSTFIELDS);  
6. curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
7. curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);  
8. curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);  
9. curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);  
10. curl_setopt($ch, CURLOPT_REFERER, true);  
11. curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);  
12. curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);  
13. $login_result = curl_exec ($ch);  
14. curl_close ($ch);  
15.  
16. $ch = curl_init('page url');  
17. curl_setopt($ch, CURLOPT_HEADER, false);  
18. curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
19. curl_setopt($ch, CURLOPT_FAILONERROR, true);  
20. curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie_file_path);  
21. curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie_file_path);  
22. $resp = curl_exec($ch);  
23. curl_close($ch);  
24.  
25. unlink($cookie_file_path);
```

Κώδικας 4.2.9: Προσκόμιση περιεχομένων σελίδας που απαιτεί σύνδεση

Περιγραφή των ετικετών (tags)

Tag == *schedule*

Στην περίπτωση κατά την οποία η ετικέτα που δέχεται από την εφαρμογή είναι το *schedule*, τότε πρέπει να επιστραφεί το εβδομαδιαίο πρόγραμμα προπτυχιακών. Για την επίτευξη αυτού απαιτούνται οι εξής τρεις ενέργειες:

- Σύνδεση στον ιστότοπο της σχολής με του προσωπικούς κωδικούς του χρήστη.
- Προσπέλαση της ιστοσελίδας που περιέχει το πρόγραμμα και επιστροφή των περιεχομένων της ως αλφαριθμητικό.
- Επεξεργασία των δεδομένων και επιστροφή τους στην εφαρμογή ως JSON πίνακα.

Τα δύο πρώτα βήματα περιγράφονται στην ενότητα «Σύνδεση στον ιστότοπο της σχολής και το *studentWeb*» της προηγούμενης σελίδας, ενώ το τρίτο βήμα περιγράφεται στον κώδικα 4.2.10. Το πρόγραμμα κάθε ημέρας βρίσκεται στις κλάσεις με όνομα “*tab_content*”, ενώ κάθε μάθημα αποτελεί μία νέα γραμμή (στοιχείο *tr*). Τα δεδομένα επεξεργάζονται στον εξυπηρέτη και επιστρέφονται στην εφαρμογή στην μορφή ενός JSON πίνακα, ο οποίος αποτελείται από 5 υποπίνακες. Κάθε υποπίνακας περιέχει αντικείμενα με 5 πεδία το καθένα (*ora*, *mathima*, κ.α.).

```
1. $dom = new DOMDocument();
2. @$dom->loadHTML($resp);
3. $xpath = new DOMXPath($dom);
4. $tabs = $xpath->query('//div[@class="tab_content"]');
5. $imera_array = array();
6. $temp_array = array();
7.
8. for($i = 0; $i < $tabs->length; $i++){
9.     $schedule_resp = array();
10.    $tr = $tabs->item($i)->getElementsByTagName('tr');
11.    for($j = 0; $j < $tr->length; $j++){
12.        if($j == 0){
13.            continue;
14.        }
15.        $td = $tr->item($j)->getElementsByTagName('td');
16.        $temp_array["ora"] = $td->item(0)->nodeValue;
17.        $temp_array["mathima"] = $td->item(1)->c14n();
18.        $temp_array["typos"] = $td->item(2)->nodeValue;
19.        $temp_array["aithousa"] = $td->item(3)->nodeValue;
20.        $temp_array["didaskwn"] = $td->item(4)->c14n();
21.
22.        array_push($schedule_resp, $temp_array);
23.    }
24.    $imera_array[(string)$i] = $schedule_resp;
25. }
26. echo json_encode($imera_array, JSON_UNESCAPED_UNICODE);
```

Κώδικας 4.2.10: Επεξεργασία δεδομένων του εβδομαδιαίου προγράμματος προπτυχιακών

Tag == login

Η ετικέτα αυτή αποστέλλεται κατά την σύνδεση του χρήστη στην εφαρμογή. Οι παρακάτω διαδικασίες εκτελούνται κατά τη σύνδεση του χρήστη:

- Αρχικά, πραγματοποιείται LDAP authentication (Κώδικας 4.2.11) για να επιβεβαιωθεί εάν ο χρήστης αποτελεί φοιτητή του τμήματος. Στην περίπτωση επιτυχούς επιβεβαίωσης, πραγματοποιείται σύνδεση στον ιστότοπο της σχολής και προσκόμιση των σελίδων του προγράμματος και των προπτυχιακών ανακοινώσεων. Σε αντίθετη περίπτωση, επιστρέφεται μήνυμα σφάλματος. Τα περιεχόμενα του εβδομαδιαίου προγράμματος επιστρέφονται, όπως περιγράφηκε στην ενότητα “Tag == schedule”, ενώ τα περιεχόμενα των προπτυχιακών ανακοινώσεων επιστρέφονται ως αφηρημένο και η επεξεργασία τους λαμβάνει χώρα στην εφαρμογή.

```
1. $ldaprdn = 'uid=' . $username . ',ou=people,dc=uth,dc=gr'
2. $ldappass = $password; // associated password
3. // connect to ldap server
4. $ldapconn = ldap_connect("ldap.uth.gr")
5.     or die("Could not connect to LDAP server.");
6.
7. // Set some ldap options for talking to
8. ldap_set_option($ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);
9. ldap_set_option($ldapconn, LDAP_OPT_REFERRALS, 0);
10.
11. if ($ldapconn) {
12.     // binding to ldap server
13.     $ldapbind = @ldap_bind($ldapconn, $ldaprdn, $ldappass);
14.     // verify binding
15.     if ($ldapbind) {
16.         // "LDAP bind successful...";
17.     }
18.     else{
19.         // "LDAP bind failed...";
20.     }
21. }
```

Κώδικας 4.2.11: LDAP authentication

Tag == grades

Στην ετικέτα grades υλοποιείται η προσκόμιση των προσωπικών βαθμών του χρήστη. Αφού πραγματοποιηθεί σύνδεση στο studentWeb και προσκόμιση των περιεχομένων της σελίδας των βαθμών, δημιουργείται ένας JSON πίνακας. Αποτελείται από αντικείμενα με 4 πεδία το καθένα (eksamino, mathima, κ.α.) και επιστρέφεται στην εφαρμογή (Κώδικας 4.2.12).

```

1. $tr = $dom->getElementsByTagName('tr');
2.
3. for($i = 0; $i < $tr->length; $i++){
4.     if($tr->item($i)->getAttribute('class') == 'italicHeader'){
5.         $eksamino++;
6.         continue;
7.     }
8.     if($tr->item($i)->hasAttribute('bgcolor')){
9.         $td = $tr->item($i)->getElementsByTagName('td');
10.        $temp_array["eksamino"] = $eksamino;
11.        $temp_array["mathima"] = $td->item(1)->nodeValue;
12.        $temp_array["typos"] = $td->item(2)->nodeValue;
13.        $temp_array["vathmos"] = $td->item(6)->nodeValue;
14.
15.        array_push($grades_resp, $temp_array);
16.        continue;
17.    }
18. }
19. echo json_encode($grades_resp, JSON_UNESCAPED_UNICODE);

```

Κώδικας 4.2.12: Επεξεργασία των βαθμών του χρήστη

Tag == announcement & tag == genikes_ann

Στην ετικέτα αυτή επιστρέφονται οι προπτυχιακές και οι γενικές ανακοινώσεις τμήματος αντίστοιχα ως αλφαριθμητικά. Η μόνη διαφορά τους είναι ότι για τις προπτυχιακές ανακοινώσεις απαιτείται η σύνδεση στον ιστότοπο του τμήματος με τους κωδικούς χρήστη.

Tag == announcement_find

Για την αναπαράσταση της πλήρους μορφής μιας προπτυχιακής ανακοίνωσης (AnakItemView.java) στην εφαρμογή, απαιτείται η αποστολή των κωδικών του χρήστη, ώστε να προσκομιστεί το περιεχόμενό της. Στη συγκεκριμένη ετικέτα, πέρα των κωδικών χρήστη ως παραμέτρων, λαμβάνεται και η παράμετρος με όνομα “link”, η οποία περιέχει τον υπερσύνδεσμο της ανακοίνωσης. Στην συνέχεια οι επόμενες δύο ενέργειες λαμβάνουν χώρα:

- Σύνδεση στον ιστότοπο της σχολής με του προσωπικούς κωδικούς του χρήστη.
- Προσπέλαση της ιστοσελίδας που περιέχει την εν λόγω ανακοίνωση και επιστροφή των περιεχομένων της ως αλφαριθμητικό στην εφαρμογή.

Tag == bus

Η ετικέτα αυτή είναι υπεύθυνη για την προσκόμιση, την επεξεργασία και την αποστολή των δρομολογίων των αστικών λεωφορείων του δήμου Βόλου. Τα δεδομένα που επιστρέφονται στην εφαρμογή αποτελούνται από ένα JSON πίνακα με αντικείμενα, όπου το καθένα περιέχει το νούμερο διαδρομής, το αρχικό δρομολόγιο και το δρομολόγιο επιστροφής.

```
1. $tr = $dom->getElementsByTagName('tr');
2.
3. for($i = 0; $i < $tr->length; $i++){
4.     if(strpos($tr->item($i)->nodeValue, "No") === false){
5.         continue;
6.     }
7.     $td = $tr->item($i)->getElementsByTagName('td');
8.     $temp_array["no"] = $td->item(0)->nodeValue;
9.     $temp_array["dromologio1"] = $td->item(1)->nodeValue;
10.    $i++;
11.    $td = $tr->item($i)->getElementsByTagName('td');
12.    $temp_array["dromologio2"] = trim($td->item(0)->nodeValue);
13.    array_push($buses_resp, $temp_array);
14. }
```

Κώδικας 4.2.13: Επεξεργασία των δρομολογίων των αστικών λεωφορείων

Tag == bus_line_find

Κατά λήψη αυτής της ετικέτας λαμβάνονται επίσης το δρομολόγιο, καθώς και η θέση του στο listview της εφαρμογής. Μετά από επεξεργασία, στην εφαρμογή επιστρέφεται ο υπερσύνδεσμος με το Pdf που παρουσιάζει το συγκεκριμένο δρομολόγιο.

```
1. $find_url = 'http://astikovoulou.gr/' . trim($item->getElementsByTagName('a')->item($position)->getAttribute('href'));
```

Κώδικας 4.2.14: Url με το pdf του δρομολογίου

Tag == menu_find

Τα δεδομένα που επιστρέφονται είναι ένα κείμενο και οι δύο υπερσύνδεσμοι, για το παλιό και το καινούργιο πρόγραμμα σίτισης.

```
1. $tab = $xpath->query('//div[@id="content80"]');
2. $p = $tab->item(0)->getElementsByTagName('p');
3. $a = $tab->item(0)->getElementsByTagName('a');
4. $lesxi_array["keimeno"] = $p->item(2)->nodeValue . "\n\n" . $p->item(3)
5. ->nodeValue . "\n\n" . $p->item(4)->nodeValue . "\n\n" ;
```

```

6. $lesxi_array["link1"] = "http://uth.gr" . $a->item(0)
7. ->getAttribute('href');
8. $lesxi_array["link2"] = "http://uth.gr" . $a->item(1)
9. ->getAttribute('href');

```

Κώδικας 4.2.15: Πρόγραμμα σίτισης

Tag == fisiognomia

Επιστρέφονται τα περιεχόμενα της σελίδας που περιέχει τη φυσιογνωμία του τμήματος ως αλφαριθμητικό.

Tag == courses

Σε αυτή την ετικέτα επιστρέφονται οι τίτλοι των προπτυχιακών μαθημάτων ως αντικείμενα ενός JSON πίνακα.

```

1. $tabs = $xpath->query('//div[@id="tabs-1-1"]');
2. $h3 = $tabs->item(0)->getElementsByTagName('h3');
3.
4. for($i = 0; $i < $h3->length; $i++){
5.     $temp_array["mathima"] = $h3->item($i)->nodeValue;
6.     array_push($courses_resp, $temp_array);
7. }

```

Κώδικας 4.2.16: Προπτυχιακά μαθήματα

Tag == didaskontes

Τα αντικείμενα του JSON πίνακα που επιστρέφεται περιέχουν τρία πεδία, το όνομα του διδάσκοντα, την κατηγορία στην οποία ανήκει και ο υπερσύνδεσμος που περιέχει το προφίλ του (Κώδικας 4.2.17).

```

1. $cat = $dom->getElementsByTagName('h6');
2. $name = $dom->getElementsByTagName('h3');
3. $tabs = $xpath->query('//div[@class="pf-content"]');
4. $m = 0;
5. for($i = 0; $i < $cat->length; $i++){
6.     $items = $tabs->item($i+1)->getElementsByTagName('tbody');
7.     for($j = 0; $j < $items->length; $j++){
8.         $trs = $items->item($j)->getElementsByTagName('tr');
9.
10.         for($k = 0; $k < $trs->length; $k++){
11.

```

```

12.         $prof = $trs->item($k-1);
13.         $temp_array["onoma"] = $name->item($m)->nodeValue;
14.         $temp_array["katigoria"] = $cat->item($i)->nodeValue;
15.         $temp_array["profil"] = $prof->c14n();
16.         array_push($didask_resp, $temp_array);
17.         $m++;
18.     }
19. }

```

Κώδικας 4.2.17: Διδάσκοντες

Tag == aithousa_find && tag == map

Και οι δύο αυτές ετικέτες πραγματοποιούν σύνδεση με τη βάση δεδομένων, η οποία περιέχει τις συντεταγμένες διάφορων σημείων (Κώδικας 4.2.18). Η διαφορά τους είναι ότι στη πρώτη ετικέτα επιλέγεται μία συγκεκριμένη τοποθεσία από την βάση (Κώδικας 4.2.19), ενώ στη δεύτερη επιλέγεται όλη η βάση (Κώδικας 4.2.20) και επιστρέφεται στην εφαρμογή ως ένα αλφαριθμητικό.

```

1. $conn = new mysqli('localhost', $user, $pass, $db) or die('Unable to connect');
2. mysqli_query($conn, "SET NAMES 'utf8'");

```

Κώδικας 4.2.18: Σύνδεση στη βάση δεδομένων map_db

```

1. $aithousa = $_POST['aithousa'];
2. $result = mysqli_query($conn, "SELECT lat , lng FROM map_table WHERE name = '$aithousa' ");

```

Κώδικας 4.2.19: Επιλογή συγκεκριμένης τοποθεσίας

```

1. $result = mysqli_query($conn, "SELECT * FROM map_table");

```

Κώδικας 4.2.20: Επιλογή ολόκληρης της βάσης

Tag == dilosi

Κατά την λήψη της ετικέτας “dilosi” λαμβάνονται και οι κωδικοί του χρήστη, ώστε να πραγματοποιηθεί σύνδεση στο studentWeb και να προσκομιστεί η δήλωσή του. Στην συνέχεια επεξεργάζονται τα δεδομένα και στην εφαρμογή επιστρέφεται ένας JSON πίνακας με αντικείμενα, τα οποία αποτελούν τους κωδικούς των μαθημάτων που έχει δηλώσει ο χρήστης στο τρέχον εξάμηνο.


```

1. $tds = $xpath->query('//td[@nowrap="true"]');
2.
3. for($i = 0; $i < $tds->length; $i++){
4.     $math = $tds->item($i)->textContent;
5.     if(strpos($math, '(') !== false){
6.         $first = strpos($math, '(');
7.         $temp_array["mathima"] = substr($math, $first);
8.         array_push($dilosi_resp, $temp_array);
9.     }
10. }
11. echo json_encode($dilosi_resp, JSON_UNESCAPED_UNICODE);

```

Κώδικας 4.2.21: Επεξεργασία δήλωσης μαθημάτων του χρήστη

Tag == course_find

Με την ετικέτα αυτή λαμβάνεται και ο κωδικός του μαθήματος με όνομα παραμέτρου “course”. Στην συνέχεια πραγματοποιείται σύνδεση στη βάση δεδομένων των μαθημάτων και επιλέγεται το `page_id` από τον συγκεκριμένο κωδικό μαθήματος (Κώδικας 4.2.22). Αφού προσκομιστούν τα περιεχόμενα της σελίδας του εν λόγω μαθήματος, επιστρέφονται στην εφαρμογή. Η επεξεργασία τους λαμβάνει χώρα στην εφαρμογή.

```

1. $conn = new mysqli('localhost', $user, $pass, $db) or die('Unable to connect');
2. mysqli_query($conn, "SET NAMES 'utf8'");
3. $result = mysqli_query($conn, "SELECT page_id FROM coursesIDs_table WHERE sult = mysqli_query($conn, "SELECT page_id FROM coursesIDs_table WHERE course_id = '$course' ");
4.
5. if(mysqli_num_rows($result)){
6.     $course_url = "http://www.inf.uth.gr/?page_id=" . mysqli_fetch_object($result)->page_id;
7. }

```

Κώδικας 4.2.22: Εύρεση url μαθήματος με τη βοήθεια της βάσης δεδομένων των προπτυχιακών μαθημάτων

5 Διεπαφή χρήστη (UI) εφαρμογής

5.1 Εικονίδιο εφαρμογής

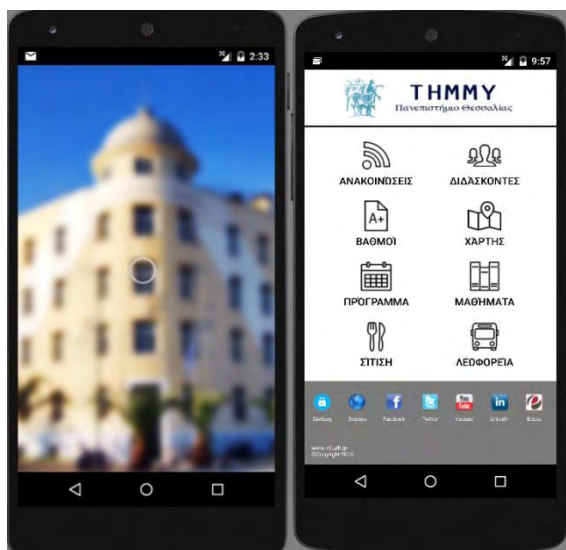
Η εικόνα 5.1.1 αποτελεί το εικονίδιο της εφαρμογής.



Εικόνα 5.1.1: Εικονίδιο εφαρμογής

5.2 Εικόνα εισόδου & αρχική σελίδα της εφαρμογής

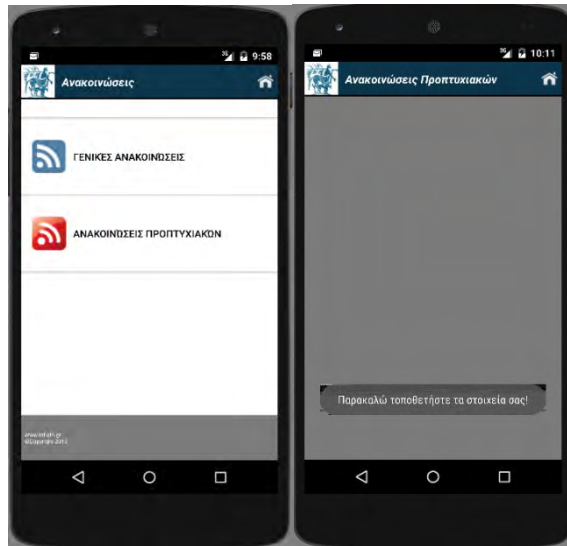
Η εικόνα 5.2.1 παρουσιάζει την εικόνα εκκίνησης της εφαρμογής και την αρχική της σελίδα. Τα μεγάλα εικονίδια αποτελούν τα κύρια activities. Στα μικρά εικονίδια υπάρχει η δυνατότητα οριζόντιας κύλισης.



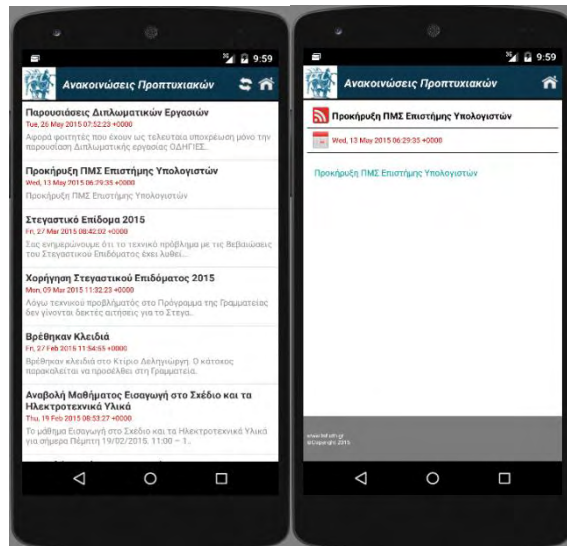
Εικόνα 5.2.1: Εικόνα εκκίνησης & αρχική σελίδα

5.3 Ανακοινώσεις

Επιλέγοντας ένα από τα δύο κουμπιά ο χρήστης κατευθύνεται στο αντίστοιχο activity. Για τη εμφάνιση των προπτυχιακών ανακοινώσεων απαιτείται η σύνδεσή του στην εφαρμογή (Εικόνα 5.3.1).



Εικόνα 5.3.1: Επιλογή τύπου ανακοινώσεων (αριστερά), μήνυμα μη συνδεσιμότητας χρήστη (δεξιά)



Εικόνα 5.3.2: Λίστα ανακοινώσεων (αριστερά), πλήρης ανακοίνωση (δεξιά)

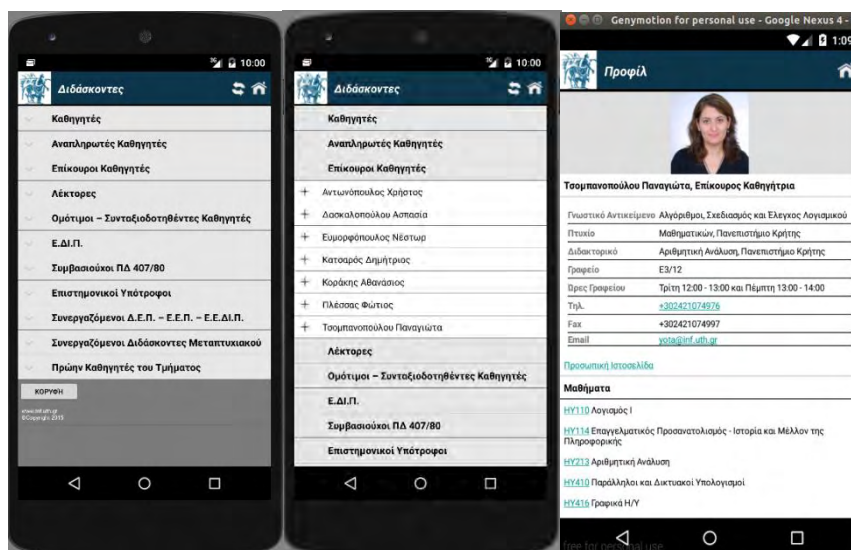
Στην εικόνα 5.3.2 παρουσιάζεται η διεπαφή χρήστη (UI) της λίστας των ανακοινώσεων και η πλήρης μορφή μίας ανακοίνωσης (είτε γενικών, είτε προπτυχιακών). Υπάρχει η δυνατότητα ανανέωσης των περιεχομένων και ανακατεύθυνσης στην αρχική σελίδα της εφαρμογής πατώντας πάνω στα αντίστοιχα κουμπιά.

5.4 Διδάσκοντες

Στην εικόνα 5.4.1 παρουσιάζεται η διεπαφή χρήστη των διδασκόντων και το προφίλ τους. Αριστερά φαίνονται οι κατηγορίες, στις οποίες ανήκουν, ενώ στο κέντρο φαίνονται οι διδάσκοντες από μία κατηγορία, μετά το πάτημα στο αντίστοιχο αντικείμενο στο ExpandableListView. Με την επιλογή ενός διδάσκοντα, γίνεται μετάβαση στο προφίλ του (Δεξιά). Οι παρακάτω ενέργειες είναι διαθέσιμες στο προφίλ του διδάσκοντα:

- Κλήση στο τηλέφωνο του γραφείου του.
- Αποστολή email.
- Επίσκεψη στην προσωπική του σελίδα.
- Ανακατεύθυνση στα μαθήματα που είναι υπεύθυνος μέσα στην εφαρμογή.

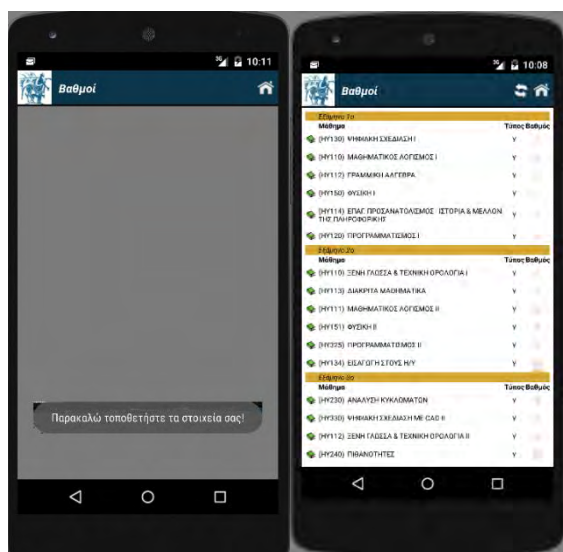
Τέλος, υπάρχει δυνατότητα ανανέωσης των περιεχομένων και ανακατεύθυνσης στην αρχική σελίδα της εφαρμογής.



Εικόνα 5.4.1: Λίστα διδασκόντων (αριστερά, κέντρο), προφίλ διδάσκοντα (δεξιά)

5.5 Βαθμοί

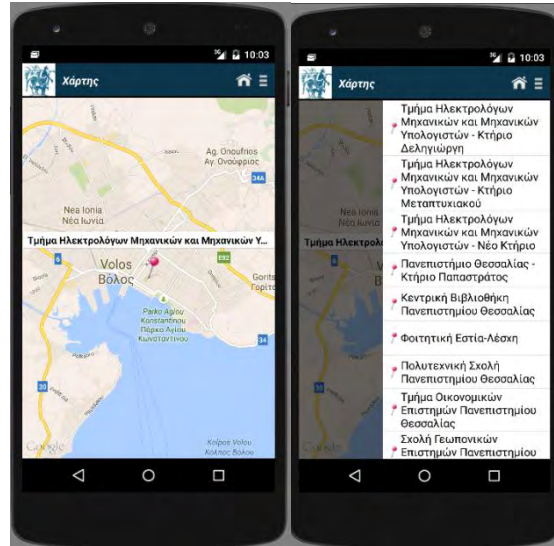
Στην εικόνα 5.5.1 παρουσιάζονται οι προσωπικοί βαθμοί του χρήστη. Απαιτείται σύνδεση στην εφαρμογή για την προβολή τους (Αριστερά). Πατώντας στο κουμπί της ανανέωσης πραγματοποιείται ανανέωση της προβολής και των περιεχομένων από το studentWeb. Τέλος, με το home κουμπί ο χρήστης επιστρέφει στην αρχική σελίδα.



Εικόνα 5.5.1: Προσωπικοί βαθμοί του χρήστη

5.6 Χάρτης

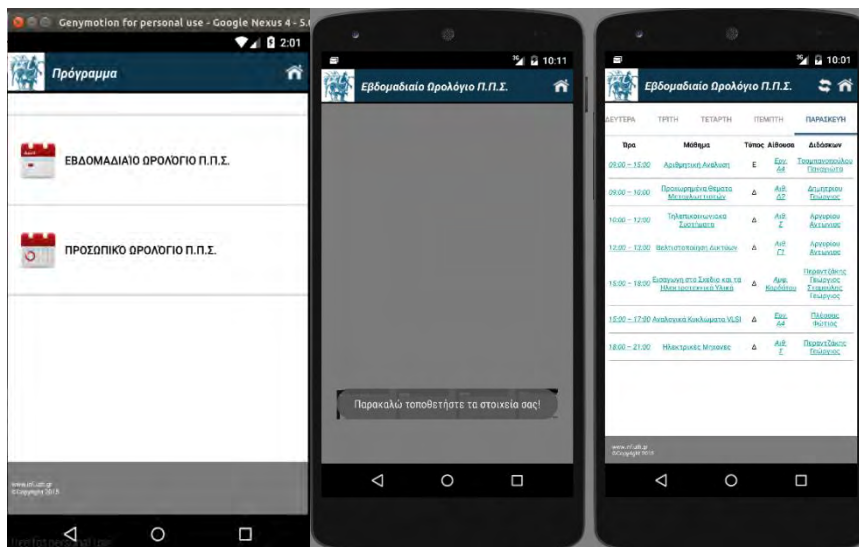
Στο activity του χάρτη ο χρήστης έχει την δυνατότητα εξερεύνησης χώρων του πανεπιστημίου και άλλων σημείων στην πόλη του Βόλου. Στην εικόνα 5.6.1 φαίνεται η παρουσίαση ενός σημείου πάνω στο χάρτη. Με το πάτημα του δεξιού κουμπιού (toggle button) ή σύροντας το δάχτυλο του από την δεξιά πλευρά του κινητού και προς τα αριστερά, ο χρήστης μπορεί να επιλέξει οποιοδήποτε σημείο επιθυμεί από την εμφανιζόμενη λίστα (Δεξιά). Τέλος, με το πάτημα του home κουμπιού πραγματοποιείται ανακατεύθυνση στην αρχική σελίδα.



Εικόνα 5.6.1: Χάρτης

5.7 Πρόγραμμα

Η εικόνα 5.7.1 παρουσιάζει το εβδομαδιαίο πρόγραμμα του τμήματος.

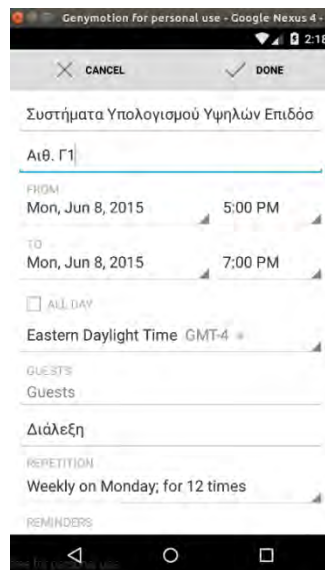


Εικόνα 5.7.1: Εβδομαδιαίο Π.Π.Σ

Αριστερά φαίνεται η δυνατότητα επιλογής ανάμεσα στο εβδομαδιαίο ωρολόγιο Π.Π.Σ του τμήματος ή του προσωπικού ωρολογίου Π.Π.Σ του χρήστη με βάση τη δήλωση μαθημάτων που έχει πραγματοποιήσει κατά το τρέχον εξάμηνο. Στο κέντρο εμφανίζεται μήνυμα σφάλματος, εάν ο χρήστης δεν έχει συνδεθεί στην εφαρμογή. Τέλος, δεξιά φαίνεται η διεπαφή χρήστη του εβδομαδιαίου Π.Π.Σ, η οποία είναι ίδια με το προσωπικό ωρολόγιο Π.Π.Σ. Ο χρήστης έχει την δυνατότητα:

- να δημιουργήσει ένα γεγονός στο ημερολόγιο του τηλεφώνου του, πατώντας πάνω στην ώρα του μαθήματος (Εικόνα 5.7.1). Στο γεγονός ορίζονται ως προεπιλογές:
 - ο τίτλος του μαθήματος.
 - η αίθουσα διδασκαλίας που λαμβάνει χώρα το μάθημα.
 - η ώρα έναρξης και η ώρα λήξης του μαθήματος.
 - διάλεξη ή εργαστήριο, ανάλογα τον τύπο του μαθήματος.
 - υπενθύμιση για 12 εβδομάδες.
- να δει το προφίλ του μαθήματος, πατώντας πάνω στο τίτλο του.
- να βρει την αίθουσα διδασκαλίας πάνω στο χάρτη, πατώντας πάνω της.
- να δει το προφίλ του διδάσκοντα, πατώντας πάνω στο όνομά του.

Τέλος, υπάρχει δυνατότητα ανανέωσης των περιεχομένων και ανακατεύθυνσης στην αρχική σελίδα της εφαρμογής.



Εικόνα 5.7.2: Δημιουργία συμβάντος στο ημερολόγιο του χρήστη για το μάθημα «Συστήματα Υπολογισμού Υψηλών Επιδόσεων»

5.8 Μαθήματα

Στην εικόνα 5.8.1 φαίνονται οι κατηγορίες των μαθημάτων και τα μαθήματα σε κάθε κατηγορία. Προσφέρεται η δυνατότητα ανανέωσης και επιστροφής στην αρχική σελίδα.



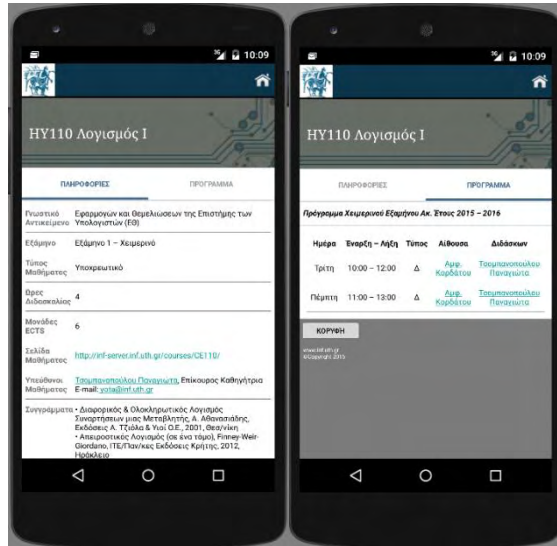
Εικόνα 5.8.1: Προπτυχιακά μαθήματα

Στην εικόνα 5.8.2 παρουσιάζεται το προφίλ ενός μαθήματος. Υπάρχουν δύο σελίδες, στις οποίες μπορεί να μεταβεί ο χρήστης. Στις γενικές πληροφορίες για το μάθημα, υπάρχει η δυνατότητα:

- μετάβασης στη σελίδα του μαθήματος στον εξυπηρέτη της σχολής.
- μετάβασης στο προφίλ του διδάσκοντα μέσα στην εφαρμογή.
- αποστολής email στο διδάσκοντα.

Στο πρόγραμμα του μαθήματος ο χρήστης μπορεί να δει την αίθουσα διδασκαλίας πάνω στο χάρτη και να μεταβεί στο προφίλ του διδάσκοντα μέσα στην εφαρμογή.

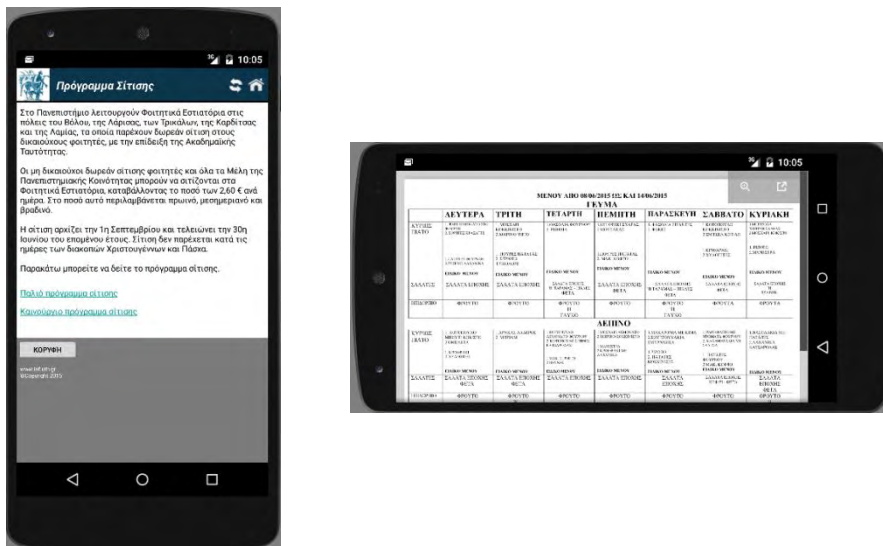
Η επιστροφή στην αρχική σελίδα πραγματοποιείται με το πάτημα του home κουμπιού.



Εικόνα 5.8.1: Προφίλ μαθήματος «Λογισμός Ι»

5.9 Σίτιση

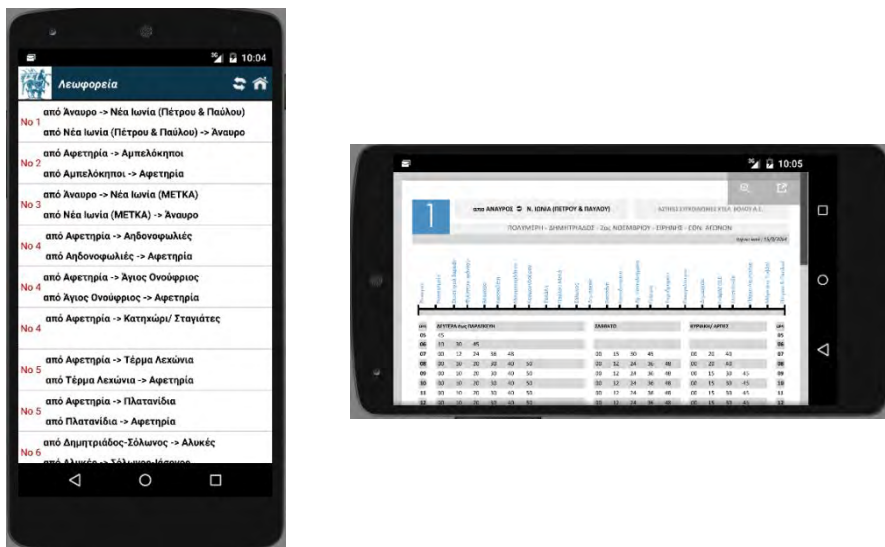
Πατώντας πάνω στους δύο συνδέσμους ο χρήστης μπορεί να δει το πρόγραμμα σίτισης (Εικόνα 5.9.1).



Εικόνα 5.9.1: Πρόγραμμα σίτισης

5.10 Λεωφορεία

Στην εικόνα 5.10.1 παρουσιάζονται τα δρομολόγια των αστικών λεωφορείων του δήμου Βόλου. Ο χρήστης μπορεί να επιλέξει και δει, όποιο δρομολόγιο επιθυμεί, πατώντας πάνω του. Μπορεί επίσης να ανανεώσει τα περιεχόμενα και να επιστρέψει στην αρχική σελίδα πατώντας στα αντίστοιχα κουμπιά.



Εικόνα 5.10.1: Λίστα δρομολογίων αστικών λεωφορείων (αριστερά), εμφάνιση δρομολογίου (δεξιά)

5.11 Φυσιογνωμία

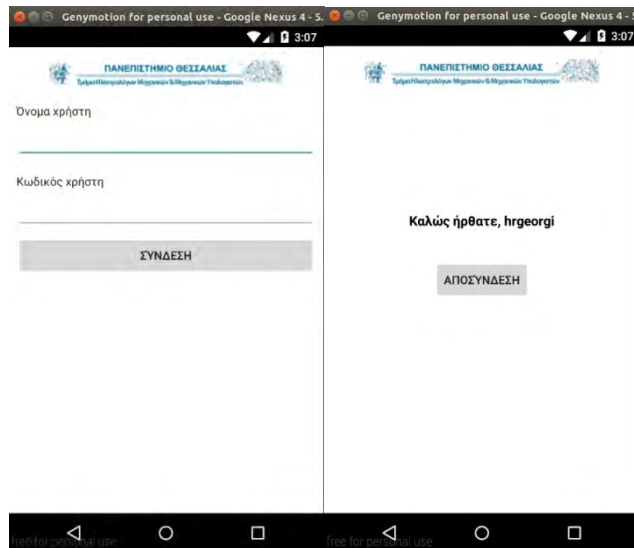
Στην εικόνα 5.11.1 της επόμενης σελίδας παρουσιάζεται η φυσιογνωμία του τμήματος.

5.12 Σύνδεση

Στην εικόνα 5.12.1 της επόμενης σελίδας ο χρήστης έχει την δυνατότητα να συνδεθεί στην εφαρμογή συμπληρώνοντας τα στοιχεία του και πατώντας στο κουμπί «Σύνδεση». Εάν τα στοιχεία που έχει τοποθετήσει είναι σωστά εμφανίζεται η δεξιά εικόνα. Μετά από 1 δευτερόλεπτο πραγματοποιείται ανακατεύθυνση στην αρχική σελίδα της εφαρμογής. Στην περίπτωση που είναι ήδη συνδεδεμένος, πατώντας στο κουμπί «Αποσύνδεση» αποσυνδέεται από την εφαρμογή.



Εικόνα 5.11.1: Φυσιογνωμία τμήματος



Εικόνα 5.12.1: Σύνδεση και αποσύνδεση του χρήστη

6 Ανάπτυξη εφαρμογής, απαιτήσεις συστήματος & δοκιμές

6.1 Ανάπτυξη εφαρμογής

Android εφαρμογή

Η ανάπτυξη του κώδικα της εφαρμογής πραγματοποιήθηκε με το Android Studio 1.2.1.1^[42]. Για τη δοκιμή της εφαρμογής κατά την διάρκεια της ανάπτυξής της χρησιμοποιήθηκε ένας εξομοιωτής (emulator) με τη βοήθεια του Genymotion^[43]. Πιο συγκεκριμένα, τα χαρακτηριστικά της συσκευής που δημιουργήθηκε παρουσιάζονται στον πίνακα 6.1.1.

Όνομα συσκευής	LG Nexus 4
Έκδοση λογισμικού (Android OS)	v5.0 (Lollipop)
Μέγεθος και ανάλυση οθόνης	4.7 inches, 768x1280 pixels
Επεξεργαστής	Qualcomm Snapdragon S4 Pro, Quad-core 1.5 GHz Krait
Μνήμη	2 GB RAM

Πίνακας 6.1.1: Χαρακτηριστικά emulator

Εξυπηρέτης

Για την υλοποίηση του εξυπηρέτη χρησιμοποιήθηκε το XAMPP 5.5.24-0 για Linux. Τα χαρακτηριστικά του συστήματος, στο οποίο φιλοξενήθηκε φαίνονται στον πίνακα 6.1.2 που ακολουθεί.

Έκδοση λειτουργικού	Linux OS, Ubuntu 15.04 (vivid)
GNOME	3.14.1
Kernel	3.19.0-18-generic
Επεξεργαστής	Intel® Xeon® W350 @ 3.07GHz
Αριθμός επεξεργαστών	1
Μνήμη	2136 MB

Πίνακας 6.1.2: Χαρακτηριστικά συστήματος, στο οποίο φιλοξενήθηκε ο server

6.2 Απαιτήσεις συστήματος

Για να είναι σε θέση ο χρήστης να εγκαταστήσει την εφαρμογή, θα πρέπει να έχει στην κατοχή του συσκευή με έκδοση Android 4.0.3 (Ice Cream Sandwich) ή μεταγενέστερη. Κατά την εγκατάστασή της, ο χρήστης ενημερώνεται για την πρόσβαση της εφαρμογής στα εξής δικαιώματα:

- ακριβής θέση (GPS και βάσει δικτύου)
- κατά προσέγγιση θέση (βάσει δικτύου)
- ανάγνωση συμβάντων ημερολογίου και εμπιστευτικών πληροφοριών.

- προσθήκη ή τροποποίηση συμβάντων ημερολογίου και αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου σε προσκεκλημένους χωρίς να το γνωρίζουν οι κάτοχοι
- ανάγνωση του περιεχομένου της κάρτας SD
- τροποποίηση ή διαγραφή των περιεχομένων της κάρτας SD
- ανάγνωση διαμόρφωσης υπηρεσιών Google.
- πλήρης πρόσβαση στο δίκτυο
- προβολή συνδέσεων δικτύου

Κατά την εκκίνησή της, η εφαρμογή απαιτεί την σύνδεση της συσκευής στο Internet. Τέλος, ο χρήστης πρέπει να βρίσκεται εντός δικτύου του Πανεπιστημίου Θεσσαλίας για να είναι σε θέση να την χρησιμοποιήσει (λόγω της υλοποίησης του εξυπηρετητή με το λογισμικό XAMPP).

6.3 Δοκιμές (Beta testing)

Μετά την ολοκλήρωση της εφαρμογής, κρίθηκε σκόπιμο η δοκιμή της σε πραγματικές συσκευές, ώστε να εντοπιστούν τυχόν προβλήματα (bugs) και παραλήψεις. Το στάδιο του beta testing κύλησε σε πολύ ικανοποιητικά πλαίσια. Η εφαρμογή έμοιαζε να ανταποκρίνεται πολύ καλά, σε όποια συσκευή και αν δοκιμάστηκε. Το μόνο πρόβλημα που εντοπίστηκε ήταν ότι σε οθόνες μικρότερες των 3.5 ιντσών, δεν υπήρχε ικανοποιητική προβολή κάποιων περιεχομένων της. Εξαιτίας του πολύ μικρού αριθμού συσκευών με οθόνη μικρότερη των 3.5 ιντσών, δεν κρίθηκε αναγκαία κάποια αλλαγή στον τρόπο αναπαράστασης των περιεχομένων. Οι συσκευές που χρησιμοποιήθηκαν κατά τη φάση του beta testing, καθώς και τα χαρακτηριστικά τους εμφανίζονται στον πίνακα 6.3.1.

Όνομα συσκευής	Έκδοση Λογισμικού (Android OS)	Μέγεθος & ανάλυση οθόνης	Επεξεργαστής	Μνήμη (RAM)
Samsung I9300 Galaxy S III	v4.3 (Jelly Bean)	4.8 inches, 720x1280 pixels	Exynos 4412 Quad, Quad-Core 1.4 GHz Cortex-A9	1 GB
Samsung Galaxy Core I8260	v4.1.2 (Jelly Bean)	4.3 inches, 400x800 pixels	Qualcomm Snapdragon S4 Play, Dual-core 1.2 GHz Cortex-A5	1 GB
Samsung Galaxy Note 3	V5.0 (Lollipop)	5.7 inches, 1080x1920 pixels	Qualcomm Snapdragon 800, Quad-Core 2.3 GHz Krait 400	3 GB
LG Optimus L1 II E410	v4.1.2 (Jelly Bean)	3.0 inches, 240x320 pixels	Qualcomm Snapdragon S1, 1 GHz Cortex-A5	512 MB
LG Optimus L9 II	v4.4.2 (Kit Kat)	4.7 inches, 720x1280 pixels	Qualcomm Snapdragon 400, Dual-Core 1.4 GHz Krait	1 GB
Huawei Ascend Y530	v4.3 (Jelly Bean)	4.5 inches, 480x854 pixels	Qualcomm Snapdragon 200, Dual-Core 1.2 GHz	512 MB

Πίνακας 6.3.1: Συσκευές για beta testing

7 Επίλογος

7.1 Συμπεράσματα

Με το τέλος αυτής της προσπάθειας, ο αρχικός στόχος για τη δημιουργία μιας εφαρμογής που να καλύπτει τις ανάγκες των προπτυχιακών φοιτητών του τμήματος έχει επιτευχθεί. Ούσα μια πλήρης και λειτουργική εφαρμογή θα μπορούσε να τεθεί άμεσα στη διάθεση της φοιτητικής κοινότητας του τμήματος.

Κατά τη διάρκεια υλοποίησης της εφαρμογής, μελετήθηκε σε βάθος το λειτουργικό σύστημα Android. Η ανάγκη χρήσης εξυπηρέτη και βάσης δεδομένων έδωσε την δυνατότητα να μελετηθούν κάποιοι ευρύτεροι τομείς απαραίτητοι για την ανάπτυξη και λειτουργία αντίστοιχων εφαρμογών.

7.2 Μελλοντική επέκταση

Στόχος των εφαρμογών έξυπνων τηλεφώνων είναι η βελτίωση της εμπειρίας του χρήστη μέσω των υπηρεσιών που του προσφέρουν. Οι ανάγκες των χρηστών συνεχώς αλλάζουν, επομένως η ανάγκη για περαιτέρω βελτίωση των εφαρμογών είναι κάτι παραπάνω από επιβεβλημένη. Στην κατεύθυνση αυτή, θα μπορούσαν να υπάρξουν κάποιες επιπλέον βελτιώσεις της εφαρμογής.

Η επέκταση χρήσης της από τους μεταπτυχιακούς φοιτητές του τμήματος θα ήταν μια σημαντική βελτίωση της εφαρμογής. Επιπλέον, η χρήση του εξυπηρέτη του τμήματος θα πρόσφερε τη σημαντική δυνατότητα χρήσης της εφαρμογής και εκτός του δικτύου του πανεπιστημίου.

8 Βιβλιογραφία

- [1]. <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [2]. <http://dazeinfo.com/2014/01/23/smartphone-users-growth-mobile-internet-2014-2017/>
- [3]. <https://play.google.com/store/apps/details?id=edu.harvard.harvardmobile>
- [4]. <https://play.google.com/store/apps/details?id=gr.auth.drupal&hl=el>
- [5]. http://en.wikipedia.org/wiki/Android_%28operating_system%29
- [6]. http://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
- [7]. <http://en.wikipedia.org/wiki/XAMPP>
- [8]. <https://www.apachefriends.org/index.html>
- [9]. <http://en.wikipedia.org/wiki/OpenSSL>
- [10]. <http://en.wikipedia.org/wiki/PhpMyAdmin>
- [11]. http://en.wikipedia.org/wiki/Apache_HTTP_Server
- [12]. <http://en.wikipedia.org/wiki/PHP>
- [13]. <http://php.net/>
- [14]. <http://www.w3schools.com/php/>
- [15]. <http://en.wikipedia.org/wiki/MySQL>
- [16]. <https://www.mysql.com/>
- [17]. <http://en.wikipedia.org/wiki/HTTPS>
- [18]. http://en.wikipedia.org/wiki/Transport_Layer_Security
- [19]. <http://en.wikipedia.org/wiki/Wireshark>
- [20]. <https://www.wireshark.org/>
- [21]. http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
- [22]. <http://www.tldp.org/HOWTO/LDAP-HOWTO/authentication.html>
- [23]. http://en.wikipedia.org/wiki/Mobile_operating_system
- [24]. <http://img105.job1001.com/upload/adminnew/2015-04-03/1428013198-SVM42OZ.pdf>
- [25]. <http://developer.android.com/index.html>
- [26]. <http://jsoup.org/download>
- [27]. <https://www.youtube.com/watch?v=yhv819F44qo>
- [28]. <http://en.wikipedia.org/wiki/JSON>
- [29]. <http://stackoverflow.com/questions/14051304/send-https-post-request-to-the-server>
- [30]. <http://www.androidhive.info/>
- [31]. <http://www.android4devs.com/2015/01/how-to-make-material-design-sliding-tabs.html>

- [32]. <http://blog-emildesign.rhcloud.com/?p=403>.
- [33]. <http://blog-emildesign.rhcloud.com/?p=435>
- [34]. <http://en.wikipedia.org/wiki/CURL>
- [35]. <http://php.net/manual/en/class.domdocument.php>
- [36]. http://en.wikipedia.org/wiki/Self-signed_certificate
- [37]. <http://en.wikipedia.org/wiki/RSA>
- [38]. http://en.wikipedia.org/wiki/Triple_DES
- [39]. http://en.wikipedia.org/wiki/Data_Encryption_Standard
- [40]. <http://www.zunisoft.com/2008/10/kb-xampp-ssl-configuration-on-ubuntu.html>
- [41]. <http://shahpunyerblog.blogspot.gr/2007/10/create-self-signed-ssl-certificate-in.html>
- [42]. <https://developer.android.com/sdk/index.html>
- [43]. <https://www.genymotion.com/#/>
- [44]. <http://www.planetb.ca/syntax-highlight-word>
- [45]. <http://inf-server.inf.uth.gr/~hrgeorgi/>