



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

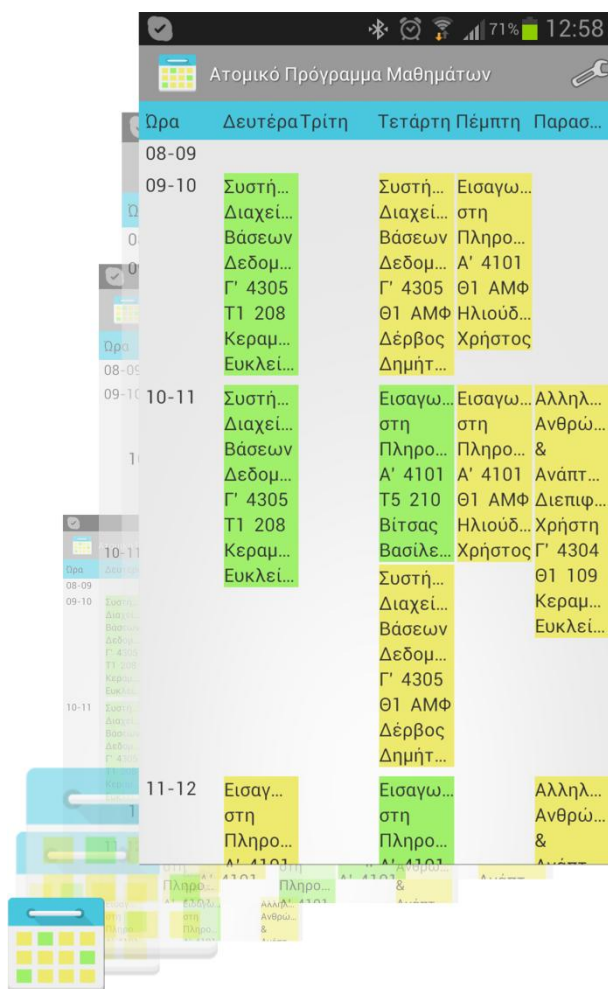


Τμήμα Μηχανικών
Πληροφορικής ΑΤΕΙΘ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή εργασία

Προβολή ατομικού προγράμματος μαθημάτων σε Android



Της φοιτήτριας

ΑΙΚΑΤΕΡΙΝΗ ΚΑΛΦΟΥΔΗ

Αρ. Μητρώου: 09/3461

Επιβλέπων καθηγητής

ΚΕΡΑΜΟΠΟΥΛΟΣ ΕΥΚΛΕΙΔΗΣ

Θεσσαλονίκη

2013

Την πτυχιακή αυτή εργασία αφιερώνω στη μνήμη του
αγαπημένου παππού μου που χάσαμε κατά τη διάρκεια εκπόνησής της.

Πρόλογος

Τα κινητά τηλέφωνα εξελίσσονται έτσι ώστε να εξυπηρετούν περισσότερο τον άνθρωπο. Πλέον δεν περιορίζονται μόνο στη επικοινωνία, αλλά παρέχουν και εφαρμογές που απασχολούν στη διευκόλυνση της καθημερινότητας.

Πρόκειται για τα λεγόμενα smartphones τα οποία τρέχουν κάποιο λειτουργικό σύστημα όπως Android, iOS, Windows, κ.α. Σε αυτά δίνεται η δυνατότητα στους προγραμματιστές να παράγουν εφαρμογές σύμφωνες με τις ανάγκες των χρηστών. Για παράδειγμα, ένα απλό ημερολόγιο μπορεί να επεκταθεί σε έναν ενημερωμένο πίνακα μαθημάτων, για να εξυπηρετήσει τους φοιτητές μιας σχολής.

Σκοπός της παρούσης πτυχιακής είναι η δημιουργία μίας εφαρμογής στο λειτουργικό σύστημα Android που θα εμφανίζει για κάθε χρήστη το προσωπικό του πρόγραμμα μαθημάτων. Θα παρουσιάζει δηλαδή τις ώρες, μέρες και αίθουσες που πρόκειται να πραγματοποιηθούν τα μαθήματα που ο κάθε φοιτητής έχει δηλώσει, καθώς και τα τμήματα και ο καθηγητής που τα διδάσκει.

Περίληψη

Η πτυχιακή αυτή εργασία παρουσιάζει την υλοποίηση μιας Android εφαρμογής που προβάλλει το ατομικό πρόγραμμα μαθημάτων ενός φοιτητή, σύμφωνα με τις επιλογές του.

Αρχικά αναφέρονται οι τεχνολογίες που χρησιμοποιούνται ξεκινώντας από το λειτουργικό σύστημα Android και την σημειογραφική γλώσσα JSON.

Στη συνέχεια περιγράφονται αναλυτικά οι απαιτήσεις της συγκεκριμένης εφαρμογής και εξηγείται η υλοποίησή της βήμα προς βήμα. Επίσης αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν για την βελτιστοποίησή της, καθώς και η δημιουργία του widget.

Τέλος σχολιάζεται και προτείνεται μία επέκτασή της με την χρήση της τεχνολογίας Near Field Communication (NFC).

Περίληψη στα Αγγλικά (Abstract)

This Thesis presents the implementation of an Android application that shows the personal schedule of a student, regarding his choices.

At first, technologies like Android and JavaScript Object Notation (JSON) Language are mention.

Later on, the requirements of this application are described and the implementation of it is explained step by step. Also, other technologies, that were used in order to optimize the application, are mentioned, as well as the widget creation.

Finally, an extension using the Near Field Communication (NFC) technology is proposed.

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τους γονείς μου που με στηρίζουν πάντα ψυχολογικά και οικονομικά στις αποφάσεις μου και στις σπουδές μου, και τον αδερφό μου που με ανέχτηκε καθ όλη τη διάρκεια προετοιμασίας αυτής της πτυχιακής. Στη συνέχεια τον καθηγητή μου, κ. Κεραμόπουλο Ευκλείδη για την καθοδήγηση και τις πολύτιμες συμβουλές που μου έδωσε. Επίσης θα ήθελα να ευχαριστήσω τον αγαπημένο μου φίλο Λουκά που μου συμπαραστέκεται διαρκώς, παρ' όλες τις δυσκολίες που αντιμετωπίζει. Τέλος τους φίλους και συναδέλφους Μπάγια Ανδρέα, Κοσματόπουλο Μιχάλη, Μένανδρο Αποστολίδη και όλους όσους μου συμπαραστάθηκαν και με βοήθησαν να φέρω εις πέρας τη συγκεκριμένη εργασία.

Πίνακας περιεχομένων

Εισαγωγή.....	11
1. Στόχος και σκοπός	11
2. Δομή της πτυχιακής εργασίας	11
ΚΕΦΑΛΑΙΟ 1: Πλατφόρμα Android.....	13
1.1 Λειτουργικό σύστημα Android.....	13
1.2 Γιατί Android;	15
1.3 Ιστορική αναδρομή, Android API levels, Ιδιαιτερότητες	15
1.4 Αρχιτεκτονική του Android	20
1.5 Δομή Android εφαρμογής	22
1.5.1 Activities.....	23
1.5.2 Services	26
1.5.3 Broadcast receivers	28
1.5.4 Content providers.....	29
1.5.5 Intents	30
1.6 Ασφάλεια και προστασία προσωπικών δεδομένων.....	31
1.7 Android και XML	33
1.7.1 AndroidManifest.xml	33
1.7.2 Layouts	36
1.7.3 Values.....	37
1.8 Google Play - Android Market	38
1.9 Το Android Bundle IDE για ανάπτυξη εφαρμογών σε Android	39
1.10 Επίλογος	39
ΚΕΦΑΛΑΙΟ 2: Τεχνολογία JSON.....	40

2.1	Τι είναι το JSON	40
2.2	Πώς, πότε και γιατί χρησιμοποιείται	43
2.3	Παράδειγμα	43
2.4	Επίλογος	44
ΚΕΦΑΛΑΙΟ 3: Η Εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων"		45
3.1	Ορισμός απαιτήσεων.....	45
3.2	Πηγή δεδομένων	46
3.3	Διάγραμμα κλάσεων	47
3.4	Σχεδίαση της εφαρμογής	50
3.5	Λειτουργικότητα προγράμματος	53
3.6	Threads - AsyncTask.....	60
3.7	Serializable - Αποθήκευση δεδομένων	61
3.8	Widget	63
3.9	Service	65
3.10	Επίλογος	65
ΚΕΦΑΛΑΙΟ 4: Τεχνολογία Near Field Communication (NFC)		66
4.1	Χαρακτηριστικά	66
4.2	Ασφάλεια	68
4.3	Εφαρμογές	69
4.4	Επέκταση της εφαρμογής με τη χρήση του NFC	70
4.5	Επίλογος	70
Ανασκόπηση - Συμπεράσματα.....		71
Αναφορές και Βιβλιογραφία.....		73
Παράρτημα 1: Πηγή δεδομένων JSON.....		76
Παράρτημα 2: Κώδικας για το Action Bar.....		104

Ευρετήριο Σχημάτων και Πινάκων

Εικόνα 1: Λογότυπο Android	13
Εικόνα 2: Η εξέλιξη του Android από την αρχή ως την έκδοση 4.4.....	17
Εικόνα 3: Στοιβά επιπέδων αρχιτεκτονικής του Android	20
Εικόνα 4: Κύκλος ζωής ενός activity	24
Εικόνα 5: Μέθοδοι διαχείρισης καταστάσεων των activities	25
Εικόνα 6: Κύκλος ζωής των services ανάλογα με την κατηγορία.....	26
Εικόνα 7: Μέθοδοι για την υλοποίηση service ανάλογα με την κατηγορία τους.....	27
Εικόνα 8: Αντιστοιχία query() με SELECT	29
Εικόνα 9: Android API levels	34
Εικόνα 10: Αναπαράσταση ενός layout από Views και ViewGroups	36
Εικόνα 11: Συνηθέστερα ViewGroups	37
Εικόνα 12: Google Play Logo	38
Εικόνα 13: Είδη τιμών της JSON	40
Εικόνα 14: Αναπαράσταση String σε JSON	41
Εικόνα 15: Αναπαράσταση αριθμού σε JSON.....	42
Εικόνα 16: Συμβολισμός αντικειμένου σε JSON.....	42
Εικόνα 17: Συμβολισμός πίνακα σε JSON.....	42
Εικόνα 18: Διάγραμμα χρηστών - στόχων	46
Εικόνα 19: Διάγραμμα κλάσεων για τα δεδομένα	48
Εικόνα 20: Εικόνα επιλογής μαθημάτων.....	50
Εικόνα 21: Action bar για την επιλογή μαθημάτων	51
Εικόνα 22: Οθόνες προβολής ατομικού προγράμματος μαθημάτων	52
Εικόνα 23: Οθόνη προβολής ατομικού προγράμματος μαθημάτων σε landscape 1 ..	53
Εικόνα 24: Οθόνη προβολής ατομικού προγράμματος μαθημάτων σε landscape 2 ..	53
Εικόνα 25: Σύμβολο loading κατά το parsing του JSON.....	55

Εικόνα 26: Βήματα εισαγωγής μαθήματος.....	55
Εικόνα 27: Περίγραμμα στα μαθήματα που πραγματοποιούνται την τρέχουσα ώρα..	57
Εικόνα 28: Scrolling	59
Εικόνα 29: Προβολή συγκεκριμένης ημέρας.....	59
Εικόνα 30: Μήνυμα τερματισμού εφαρμογής λόγω καθυστέρησης στο Main Thread.	60
Εικόνα 31: Το widget όταν το μάθημα δεν πραγματοποιείται σύντομα.....	64
Εικόνα 32: Το widget όταν πλησιάζει ή διανύεται η ώρα παρακολούθησης του μαθήματος.....	64
Εικόνα 33: Λογότυπο NFC	66
Εικόνα 34: Κεραία NFC στη μπαταρία συσκευής	67
Εικόνα 35: Ταχύτητες και εμβέλειες ασυρμάτων επικοινωνιών.....	67
Εικόνα 36: Ολοκλήρωση συναλλαγής μέσω NFC	69
Εικόνα 38: Αυτοκόλλητο NFC tag	70

Εισαγωγή

Όπως αναφέρθηκε ήδη, η εργασία αυτή περιέχει την περιγραφή τόσο της εφαρμογής που υλοποιήθηκε, όσο και των τεχνολογιών που χρησιμοποιήθηκαν. Σε αυτό το κεφάλαιο θα αναλυθεί ο στόχος της εργασίας και ο σκοπός για τον οποίο διαλέχτηκε αυτό το θέμα. Επίσης θα αναφερθούν εκτενέστερα τα κεφάλαια που πρόκειται να ακολουθήσουν.

1. Στόχος και σκοπός

Στόχος της παρούσας πτυχιακής εργασίας είναι να δημιουργηθεί μία εφαρμογή η οποία θα περιέχει το πρόγραμμα μαθημάτων του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Από το πλήρες αυτό πρόγραμμα θα μπορεί κάθε χρήστης να διαλέξει τα μαθήματα που τον ενδιαφέρει να παρακολουθεί, τα οποία και θα εμφανίζονται στην αντίστοιχη οθόνη σε εβδομαδιαία ή ημερήσια βάση. Για να γίνει όμως αυτό, πρέπει πρώτα να αναλυθούν το λειτουργικό σύστημα που θα φιλοξενεί την εφαρμογή, η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί και η σημειογραφική γλώσσα που θα περιέχει την πηγή δεδομένων.

Βασικός σκοπός της εργασίας είναι να περιγραφούν νέες τεχνολογίες που εισέρχονται στην καθημερινότητα των ανθρώπων. Επιπλέον, η εφαρμογή έχει ως σκοπό να εμπνεύσει για τη δημιουργία ενός πιο συνολικού project που θα περιλαμβάνει συγκεντρωτικά όλα τα τμήματα όλων των σχολών όλων των Ανώτερων και Ανώτατων Εκπαιδευτικών Ιδρυμάτων της χώρας.

2. Δομή της πτυχιακής εργασίας

Κάθε κεφάλαιο έχει δικό του λόγο που θα αναπτυχθεί. Αποσκοπεί δηλαδή στο να εισάγει τις βασικές έννοιες που θα χρειαστούν για την εφαρμογή που υλοποιείται, ή να περιγράψει την λειτουργία της εφαρμογής καθώς και τις τεχνολογίες που χρησιμοποιεί, ή να παρουσιάσει επεκτάσεις και βελτιώσεις. Πιο συγκεκριμένα:

Στο πρώτο κεφάλαιο θα καλυφθούν τα θέματα που σχετίζονται με το λειτουργικό σύστημα Android, όπως τα ιδιαίτερα χαρακτηριστικά του, και οι λόγοι για τους οποίους προτιμάται. Επίσης θα γίνει ιστορική αναδρομή από τη στιγμή που πρωτοεμφανίστηκε μέχρι σήμερα, ενώ θα αναφερθεί η αρχιτεκτονική του και τα βασικά συστατικά που χρησιμοποιούν οι εφαρμογές. Επιπρόσθετα, θα παρουσιαστεί η συμβολική γλώσσα

XML που χρησιμοποιείται για την δημιουργία του user interface και άλλων σημαντικών αρχείων. Τέλος, θα αναφερθούν τα ζητήματα ασφαλείας, καθώς και οι απαραίτητες ενέργειες που εκτελούνται ώστε μία εφαρμογή να γίνει διαθέσιμη στους ενδιαφερόμενους.

Στο δεύτερο κεφάλαιο θα περιγραφεί η σημειογραφική γλώσσα JSON. Θα αναπτυχθεί η σύνταξή και οι χρήσεις της, διότι θα χρειαστεί για την εισαγωγή δεδομένων της εφαρμογής.

Το τρίτο κεφάλαιο θα περιέχει την υλοποίηση της εφαρμογής. Θα αναλυθούν αρχικά οι στόχοι από τους οποίους καθορίζονται οι απαιτήσεις της. Επίσης θα εξηγηθεί η δομή του JSON που θα αποτελέσει την πηγή δεδομένων της εφαρμογής, και το διάγραμμα κλάσεων που περιέχει κάθε αντικείμενο που χρειάζεται αυτή. Θα ακολουθήσει πλήρης περιγραφή της λειτουργικότητας και των βελτιώσεων της εφαρμογής, καθώς και η δημιουργία του widget για την άμεση ενημέρωση των χρηστών.

Στο τέταρτο κεφάλαιο θα δοθεί μια σύντομη αναφορά στην τεχνολογία Near Field Communication (NFC) ώστε να εξηγηθεί ο τρόπος με τον οποίο θα μπορούσε να αποτελέσει επέκταση της εφαρμογής.

Τελευταίο κεφάλαιο θα αποτελέσουν η ανασκόπηση της εργασίας, μαζί με τα συμπεράσματα τα οποία προκύπτουν από αυτήν.

ΚΕΦΑΛΑΙΟ 1: Πλατφόρμα Android



Εικόνα 1: Λογότυπο Android

Σε αυτό το κεφάλαιο περιγράφεται το λειτουργικό σύστημα Android (Λογότυπο στην Εικόνα 1). Αρχικά αναπτύσσονται τα ιδιαίτερα χαρακτηριστικά του και οι λόγοι που έγινε τόσο δημοφιλές. Στη συνέχεια γίνεται μια ιστορική αναδρομή στις εκδόσεις του, αναφέρεται η αρχιτεκτονική του, περιγράφονται τα συστατικά που χρησιμοποιούν οι εφαρμογές του καθώς και η χρησιμότητα της συμβολικής γλώσσας XML, με παραδείγματα σχεδίασης του user interface ενός προγράμματος μέχρι τον κώδικα του, ενώ αναλύονται διάφορα θέματα ασφάλειας. Τέλος αναφέρονται η εφαρμογή Google Play με τη βοήθεια της οποίας οι εφαρμογές διατίθενται στους χρήστες και το εργαλείο Android Bundle που χρησιμεύει στην ανάπτυξη και υλοποίηση των εφαρμογών.

1.1 Λειτουργικό σύστημα Android

Το Android είναι το πρώτο ολοκληρωμένο, ελεύθερο και δωρεάν λειτουργικό σύστημα που αρχικά σχεδιάστηκε για κινητά με οθόνη αφής, όπως smartphones. Αναπτύχθηκε βασισμένο στον πυρήνα του Linux από την Android Inc. και αγοράστηκε από την Google το 2005 (Felker). Παρέχει επίσης ένα πλήρες SDK (Software Development Kit) με επαρκή εργαλεία και χαρακτηριστικά για την ανάπτυξη εφαρμογών προσφέροντας μια ολοκληρωμένη πλατφόρμα (Darcey and Conder, 2010).

Η αλληλεπίδραση του ανθρώπου με το Android γίνεται με χειρισμούς που αντιστοιχούν σε πραγματικές κινήσεις, χρησιμοποιώντας την οθόνη αφής, ή τους διάφορους αισθητήρες της συσκευής. Παραδείγματα τέτοιων κινήσεων είναι το πάτημα ενός κουμπιού που φαίνεται στην οθόνη, το σύρσιμο του δαχτύλου για να αλλάξει κάποιο tab, το πλησίασμα και η απομάκρυνση των δαχτύλων για να γίνει σμίκρυνση και μεγέθυνση αντίστοιχα. Χρησιμοποιώντας τους αισθητήρες που διαθέτουν οι συσκευές, όπως επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες εγγύτητας, μπορεί λόγω χάριν ο χρήστης να αλλάξει την προβολή της οθόνης από portrait σε landscape, απλά αλλάζοντας τον προσανατολισμό της συσκευής (Wikipedia).

Τα κινητά αυτά δίνουν τη δυνατότητα στους χρήστες να προσαρμόσουν το περιβάλλον εργασίας σύμφωνα με τις δικές τους προτιμήσεις προσθέτοντας συντομεύσεις για εφαρμογές ή και widgets τα οποία παρέχουν ενημερωμένο περιεχόμενο απ' ευθείας στην επιφάνεια εργασίας. Επιπλέον, οι εφαρμογές μπορούν να ενημερώνουν άμεσα τους χρήστες προβάλλοντας ειδοποιήσεις (Wikipedia).

Όπως προαναφέρθηκε, το Android είναι ανοιχτού κώδικα, και η Google το διανέμει σύμφωνα με το Apache License (Wikipedia). Ο πηγαίος κώδικάς του μπορεί να βρεθεί στο <http://source.android.com/> . Αυτό σημαίνει ότι οι τεχνολογικές εταιρίες ή άλλοι ιδιώτες μπορούν ελεύθερα να το τροποποιούν και να το επανακυκλοφορούν. Για τον λόγο αυτό, οι κατασκευαστές συσκευών που ψάχνουν για ένα έτοιμο, χαμηλού κόστους, προσαρμόσιμο και "ελαφρύ" λειτουργικό σύστημα διαλέγουν το Android και αφού το εμπλουτίσουν με επιπρόσθετο ιδιόκτητο λογισμικό και χαρακτηριστικά, πουλάνε τις συσκευές με την διανομή τους προεγκατεστημένη. Επίσης, πλήθος ανεξάρτητων προγραμματιστών διαθέτει εφαρμογές για το Android οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών, επειδή η κύρια γλώσσα προγραμματισμού είναι η Java (Felker).

Βασικό επίσης στοιχείο του Android είναι η συμβατότητα. Οι συσκευές όχι μόνο διαφέρουν σε χαρακτηριστικά, αλλά το λειτουργικό αυτό σύστημα είναι ανεξάρτητο από το είδος συσκευής που θα την χρησιμοποιεί. Δεν προορίζεται δηλαδή μόνο για τηλέφωνα, αλλά και για e-book readers, netbooks, συσκευές GPS, κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, ψηφιακές κορνίζες, ακόμα και για ρολόγια και τηλεοράσεις (Felker). Ως αποτέλεσμα, το Android ενσωματώνεται και σε συσκευές που κυκλοφόρησαν τρέχοντας άλλα λειτουργικά συστήματα (Wikipedia).

1.2 Γιατί Android;

Όπως προαναφέρθηκε, το Android αποτελεί μια ανοιχτή και δωρεάν πλατφόρμα, που διαθέτει η Google. Λόγω του χαμηλού κόστους των συσκευών, και του μηδενικού κόστους του λειτουργικού, έχουν ήδη πουληθεί περίπου 1 δισεκατομμύριο συσκευές (Wikipedia). Αποτελεί λοιπόν έναν εξαιρετικά εύκολο τρόπο σε νέους προγραμματιστές να μπουν στην αγορά εργασίας ως ελεύθεροι επαγγελματίες διότι μπορούν άμεσα να διαθέσουν τις εφαρμογές τους στο Google Play Store και μπορεί οι χρήστες να μην ξέρουν τους ίδιους, αλλά σίγουρα εμπιστεύονται την Google.

Επιπροσθέτως, το υλικό των συσκευών και τα APIs της πλατφόρμας δίνουν ένα πλήθος δυνατοτήτων, ενώ υπάρχουν λιγότεροι περιορισμοί σε σχέση με άλλες πλατφόρμες. Ταυτόχρονα, ένας προγραμματιστής με εμπειρία στην Java και την XML δεν θα δυσκολευτεί ιδιαίτερα να ανταπεξέλθει στις ιδιαιτερότητες της νέας αυτής τεχνολογίας (Darcey and Conder, 2010).

Άλλο σημαντικό πλεονέκτημα του Android είναι ότι τρέχει σε διαφορετικές συσκευές με διαφορετικές απαιτήσεις. Αυτό σημαίνει ότι δεν περιορίζονται οι συσκευές με χειρότερη ανάλυση, με μικρότερη οθόνη, χωρίς μπροστινή κάμερα, ή χωρίς GPS. Το Android μπορεί να λειτουργήσει και σε αυτές τις συσκευές, ενώ οι εφαρμογές διαχωρίζονται με βάση την συμβατότητα τους μέσα από το Market της Google, χωρίς να χρειάζεται περεταίρω ενέργεια από τον προγραμματιστή (Felker).

Τέλος, υπάρχει επικοινωνία μεταξύ εφαρμογών και υπηρεσιών, με τη χρήση καταλλήλων APIs, πράγμα που αυξάνει τις δυνατότητες των προγραμματιστών, χωρίς να υπάρχει ιδιαίτερη δυσκολία στην χρήση τους. Για παράδειγμα μια εφαρμογή παιχνιδιού μπορεί να χρησιμοποιεί μια βιβλιοθήκη κοινωνικής δικτύωσης, ενημερώνοντας τους φίλους του χρήστη για το επίπεδό του στο παιχνίδι. Αν επίσης αυτό συνδεθεί με το API που χειρίζεται το GPS, ο χρήστης μπορεί άμεσα να γνωρίζει ποιοι άλλοι χρήστες βρίσκονται κοντά του εκείνη τη στιγμή, ώστε να συνεργαστούν (Felker).

1.3 Ιστορική αναδρομή, Android API levels, Ιδιαιτερότητες

Η Android Inc. ιδρύθηκε από τους Andy Rubin (συνιδρυτής της Danger), Rich Miner (συνιδρυτής της Wildfire Communications, Inc.), Nick Sears (πρώην αντιπρόεδρος της

T-Mobile), και ο Chris White (επικεφαλής σχεδιαστής των διεπιφανειών στην WebTV) στην Καλιφόρνια τον Οκτώβριο του 2003. Σκοπός τους, σύμφωνα με τα λόγια του Rubin θα ήταν να υλοποιήσουν "τις εξυπνότερες κινητές συσκευές που θα είναι πιο ενήμερες για την τοποθεσία του χρήστη και τις προτιμήσεις του" (Wikipedia).

Η Google αγόρασε την Android Inc. στις 17 Αυγούστου το 2005, προσλαμβάνοντας και τους γνωστούς Rubin, Miner και White ως υπαλλήλους. Τότε, πολλοί θεώρησαν ότι η Google σκοπεύει να εισέλθει στην αγορά κινητών τηλεφώνων.

Στις 5 Νοεμβρίου του 2007 δημιουργήθηκε η Open Handset Alliance. Πρόκειται για μια κοινοπραξία εταιρειών τεχνολογίας, συμπεριλαμβανομένης της Google, των κατασκευαστών συσκευών HTC, Sony και Samsung, των κατασκευαστών πλακετών ασύρματης επικοινωνίας όπως η Sprint Nextel και η T-Mobile, και των κατασκευαστών chipset, όπως η Qualcomm και η Texas Instruments, με σκοπό να αναπτύξουν ανοιχτά πρότυπα για κινητές συσκευές. Την ίδια μέρα, η Open Handset Alliance παρουσίασε το Android ως μία ανοιχτή πλατφόρμα για κινητές συσκευές, βασισμένη στην έκδοση του πυρήνα του Linux (Wikipedia).

Το πρώτο κινητό τηλέφωνο που κυκλοφόρησε με Android ήταν το T-Mobile G1 (γνωστό ως HTC Dream για την Ευρώπη) στις 22 Οκτωβρίου του 2008 (Darcey and Conder, 2010). Τότε, πολλά από τα χαρακτηριστικά που πλέον θεωρούνται απαραίτητα στο Android, όπως το πληκτρολόγιο στην οθόνη και η δυνατότητα πολλαπλών σημείων αφής δεν υπήρχαν (The Verge). Ωστόσο, συναντάμε εξ αρχής τα εξής στοιχεία, μερικά από τα οποία επεκτάθηκαν και λειτουργούν ως σήμερα.

Αρχικά, βασικό αναγνωριστικό του Android είναι η γραμμή ειδοποιήσεων (ενημερώσεων), που περιέχει χρήσιμες πληροφορίες, όπως ώρα, σύνδεση με δίκτυα, sms μηνύματα, και επεκτείνεται όταν ο χρήστης την σύρει προς τα κάτω εμφανίζοντας αναλυτικά κάθε ειδοποίηση σε μορφή λίστας. Επίσης, από την πρώτη κιάλας έκδοση, το Android παρείχε στους χρήστες τη δυνατότητα να προσαρμόσουν στην επιφάνεια εργασίας τα widgets που προτιμούσαν, όμως το SDK δεν υποστήριζε ακόμα καμία βιβλιοθήκη για την υλοποίησή τους από εξωτερικούς προγραμματιστές (The Verge).

Το Android επιπλέον χρησιμοποιούσε εξ αρχής τα πρωτόκολλα IMAP και POP, αφού υπήρχε ήδη εφαρμογή για το Gmail, ενώ το Android Market, σε πρωταρχικό ακόμα

στάδιο, δεν περιείχε ακόμα εφαρμογές επί πληρωμή. Παρέχονταν μόνο μία φτωχή λίστα βασικών εφαρμογών (The Verge).

Η επόμενη έκδοση Android, η 1.1, ήρθε τρεις μήνες αργότερα. Δεν είχε πολλά να προσθέσει, εκτός από αρκετές διορθώσεις, όμως απέδειξε επιτυχώς στους χρήστες ότι η αναβάθμιση στην εκάστοτε έκδοση είναι μία αρκετά εύκολη και "ανώδυνη" διαδικασία. Για τους προγραμματιστές ωστόσο, κάθε αναβάθμιση συνδέεται και με νέα APIs, κάτι που σημαίνει ότι αν θέλουν οι εφαρμογές τους να τρέχουν και σε παλαιότερες εκδόσεις, πρέπει να λαμβάνουν περαιτέρω μέτρα εφ' όσον χρησιμοποιούν λειτουργίες διαθέσιμες μόνο για νεότερες εκδόσεις (The Verge).

Όλες οι μετέπειτα εκδόσεις συνοδευόντουσαν και από ένα όνομα... γλυκίσματος! Η παράδοση με τα "γλυκά" ονόματα ξεκίνησε από την έκδοση 1.5 με όνομα Cupcake, και ακολούθησαν οι Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean και, η πιο πρόσφατη, KitKat (Εικόνα 2). Φαίνεται πως η Google σκοπεύει να συνεχίσει την παράδοση αυτή για όλο το αλφάβητο (The Verge).



Εικόνα 2: Η εξέλιξη του Android από την αρχή ως την έκδοση 4.4

Στην έκδοση Cupcake 1.5, οι σημαντικότερες αλλαγές και βελτιώσεις στην λειτουργικότητα των συσκευών έρχονται με νέα χαρακτηριστικά όπως η ενσωμάτωση του εικονικού πληκτρολογίου στην οθόνη, η δυνατότητα σε προγραμματιστές να προσθέτουν και widgets στις εφαρμογές τους, μια αρχική προσπάθεια υλοποίησης του copy - paste και επέκταση της κάμερας για λήψη video. Έτσι κυκλοφόρησε το πρώτο κινητό που διέθετε μόνο οθόνη αφής, καταργώντας τα υπόλοιπα πλήκτρα (The Verge).

Κυριότερες αλλαγές στην έκδοση Donut 1.6 ήταν η υποστήριξη του πρωτοκόλλου CDMA, και η δυνατότητά του λειτουργικού να τρέχει σε συσκευές με διαφορετικές αναλύσεις οθόνης. Επιπλέον η 1.6 εισήγαγε την έννοια της "καθολικής αναζήτησης" ("universal search") μέσα από το Quick Search Box, επιτρέποντας τους χρήστες να

ψάχνουν τον τηλεφωνικό κατάλόγό τους και τις εγκατεστημένες εφαρμογές τους στο widget αναζητήσεων της Google και διαθέτοντας μεθόδους που οι προγραμματιστές μπορούν να χρησιμοποιήσουν για να φαίνονται οι εφαρμογές τους σε αυτό. Τέλος, αναδιαμορφώνεται το Android Market, παρέχοντας πλέον και τις λίστες των top δωρεάν εφαρμογών και των top εφαρμογών επί πληρωμή (The Verge).

Οι εκδόσεις Eclair (2.0 και 2.1) ήταν οι πρώτες που περιείχαν το Google Maps Navigation για πλοήγηση στους χάρτες της Google, υποστήριξη πολλαπλών λογαριασμών Google για την ίδια συσκευή ώστε να μπορεί ο χρήστης να διαχωρίζει profiles και να μπορούν οι εφαρμογές να τα χρησιμοποιούν για να συγχρονίζονται. Προστέθηκε επίσης η λειτουργία Quick Contact όπουδήποτε εμφανιζόταν το εικονίδιο κάποιας επαφής εμφανίζοντας ένα μενού επιλογών επικοινωνίας, άρχισε να χρησιμοποιείται το multitouch στο εικονικό πληκτρολόγιο για καλύτερη απόκριση, εμφανίστηκαν τα πρώτα Live Wallpapers και ενσωματώθηκε το speech-to-text ώστε να γράφεται κείμενο που ο χρήστης έχει πει (The Verge).

Με την έκδοση 2.2 Froyo βελτιώθηκε σημαντικά η ταχύτητα και η απόδοση, ενώ οι συσκευές για πρώτη φορά λειτουργούν και ως hot spots στα οποία μπορείς να συνδέσεις οποιαδήποτε άλλη συσκευή και να δει internet. Προστέθηκε επίσης και η επιλογή PIN ή κωδικού για το ξεκλείδωμα της οθόνης, αντί για ένα απλό μοτίβο (The Verge).

Η έκδοση Gingerbread 2.3 έφερε βελτιώσεις στα εργαλεία διαχείρισης της μπαταρίας και των εφαρμογών, ενώ άρχισε να υποστηρίζεται και μπροστινή κάμερα για τις συσκευές. Σημαντικότερο όμως, είναι ότι δόθηκε η δυνατότητα χειρισμού του ήχου, των γραφικών και της μνήμης της συσκευής σε χαμηλότερο επίπεδο, βοηθώντας έτσι τους προγραμματιστές να γράψουν καλύτερο κώδικα για την υλοποίηση 3D γραφικών χωρίς να κολλάει η συσκευή. Επίσης άρχισε να υποστηρίζεται και το NFC (Near Field Communication) που συνήθως λειτουργεί χάρη σε μία μικρή κεραία που βρίσκεται μέσα στην μπαταρία των συσκευών και με την χρήση διαφόρων tags μπορεί από το να αλλάζει βασικές επιλογές του κινητού, μέχρι και να πληρώνει εισιτήρια μέσω του Google Wallet (The Verge).

Το Honeycomb, έκδοση 3.0 - 3.2 ήταν η πρώτη προσπάθεια της Google να επεκταθεί και σε tablets. Δεν είχε όμως την επιτυχία που περίμενε, μολονότι έκανε κι άλλες

βελτιώσεις στην λειτουργικότητα του λογισμικού. Τα widgets εμφάνιζαν πλέον τον χώρο που πρόκειται να καταναλώσουν στην επιφάνεια εργασίας σε κουτάκια του grid, ενώ εμφανίζεται και η γραμμή ενεργειών σε κάποιες εφαρμογές, που περιέχει κουμπιά για τις πιο συχνά χρησιμοποιούμενες ενέργειες για τις εφαρμογές αυτές. Διαθέτεται προφανώς και το αντίστοιχο API για την ενσωμάτωση της γραμμής αυτής σε όποια εφαρμογή επιθυμούν οι εκάστοτε προγραμματιστές (The Verge).

Κυριότερα χαρακτηριστικά που εμφανίζονται στην έκδοση Ice Cream Sandwich 4.0, είναι ότι το NCF προεκτείνεται σε Android Beam, το οποίο μπορεί να δώσει την δυνατότητα σε δύο συσκευές που το υποστηρίζουν να ανταλλάσσουν αρχεία απλά ακουμπώντας την μία με την άλλη. Επίσης προστίθεται η επιλογή ξεκλειδώματος της συσκευής με αναγνώριση προσώπου, και μια εφαρμογή που αναλύει και απεικονίζει στατιστικά στοιχεία για την χρήση των δεδομένων που μεταφέρονται μέσω του τηλεφωνικού δικτύου (3G, 4G, κ.α.) (The Verge).

Μεγαλύτερη βάση στα γραφικά δίνει η έκδοση 4.1 Jelly Bean ενισχύοντας τριπλά τα γραφικά με αποτέλεσμα να μπορεί να επιτευχθεί επανασχεδιασμός τους κάθε 16 milliseconds. Απαλείφεται επιτέλους μια σημαντική καθυστέρηση που φαινόταν ακόμα και κατά την κύλιση της οθόνης (scroll). Προστίθεται επίσης η δυνατότητα επέκτασης των ειδοποιήσεων του πίνακα ειδοποιήσεων, ενώ τα widgets μπορούν πλέον να αλλάζουν μέγεθος από τους χρήστες, δυναμικά. Επιπλέον εμφανίζεται για πρώτη φορά στο πληκτρολόγιο μία λίστα προβλεπόμενων λέξεων με σκοπό να διευκολύνει τον χρήστη να γράφει. Μάλιστα, με ανάλογη επιλογή του χρήστη, εμφανίζονται συγκεκριμένα οι λέξεις που εκείνος χρησιμοποιεί περισσότερο, κάνοντας την λειτουργία αυτή να μαθαίνει από μόνη της (The Verge).

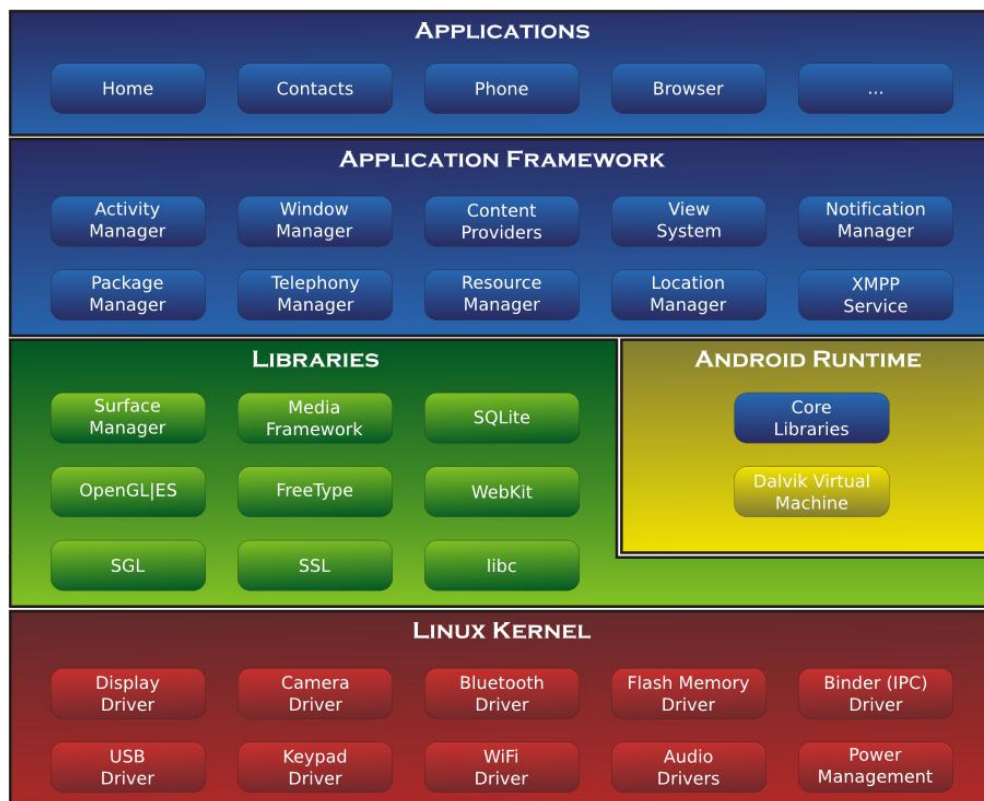
Οι μετέπειτα επανεκδόσεις του Jelly Bean (4.2 και 4.3) ενσωματώνουν στο πληκτρολόγιο το Swype με το οποίο ο χρήστης γράφει κάθε λέξη με μία συνεχόμενη κίνηση και υποστηρίζεται το Miracast με τη βοήθεια του οποίου γίνεται να συνδεθεί ασύρματα το κινητό με τηλεόραση, ή άλλη συσκευή, και να μεταδοθεί ήχος και εικόνα απ' ευθείας στην τηλεόραση. Επιπρόσθετα, εμφανίζονται τα Photospheres, τα οποία είναι πανοραμικές φωτογραφίες 360°. Υποστηρίζεται επίσης η μηχανή γραφικών OpenGL ES 3.0 για προηγμένα γραφικά παιχνιδιών (The Verge).

Τέλος, η έκδοση 4.4 Kit Kat φανερώνει την πρώτη φορά που η Google συνεργάζεται με μια εταιρία για την mascot του Android, κάνοντας έτσι μια μαζική εμπορική εκστρατεία με την Nestle. Εμφανίζονται επίσης παιχνίδια σε full screen και μια σειρά από νέα πολύχρωμα εικονίδια ενσωματωμένα στο πληκτρολόγιο (The Verge).

Η Google έχει αποδείξει ότι δεν χρειάζεται να αναβαθμιστεί ολόκληρο το λειτουργικό του Android για να γίνουν αλλαγές. Κάθε εφαρμογή χωριστά αναβαθμίζεται σε τακτά χρονικά διαστήματα παρέχοντας στους χρήστες άμεσα τις ανανεωμένες εφαρμογές, χωρίς να υπάρχει, από τους παραγωγούς, η καθυστέρηση της χρονοβόρας διαδικασίας παραμετροποίησης του λειτουργικού για της συσκευές τους (The Verge).

1.4 Αρχιτεκτονική του Android

Το λειτουργικό σύστημα Android είναι χωρισμένο σε διάφορα επίπεδα λογισμικού. Ξεκινάει από τον πυρήνα του Linux, ακολουθούν οι βιβλιοθήκες μαζί με το περιβάλλον εκτέλεσης, το framework των εφαρμογών, και καταλήγει στις εφαρμογές που παρέχονται από το σύστημα, ή από τους κατασκευαστές των συσκευών, είτε από άλλους προγραμματιστές όπως φαίνεται στην Εικόνα 3.



Εικόνα 3: Στοιβά επιπέδων αρχιτεκτονικής του Android

Στη βάση της στοίβας βρίσκεται ο πυρήνας του Linux στον οποίο βασίζεται το Android. Ποτέ δεν αλληλεπιδρά με τους χρήστες ή τους προγραμματιστές, αλλά αποτελεί το στρώμα με τους απαραίτητους οδηγούς (drivers) για την επικοινωνία με το υλικό των συσκευών ώστε να διαχειρίζεται άμεσα όλα τα υλικά κομμάτια της συσκευής, όπως την οθόνη, την κάμερα, την μνήμη, το bluetooth, το USB και το WiFi. Πρόκειται για μία απλουστευμένη έκδοση του πυρήνα του Linux, που περιέχει μόνο τα απαραίτητα χαρακτηριστικά που χρειάζεται το Android. Λόγω της ελευθερίας που παρέχει στους προγραμματιστές ως open source λογισμικό, επεκτείνεται και βελτιώνεται διαρκώς (Γκούτζος 2013).

Το επόμενο επίπεδο είναι οι native libraries, δηλαδή οι βιβλιοθήκες που είναι γραμμένες σε C και C++, δηλαδή στις κατ' εξοχήν γλώσσες του λειτουργικού. Πρόκειται για ένα σύνολο οδηγιών που έχουν σκοπό να καθοδηγήσουν την συσκευή πώς να διαχειριστεί διαφορετικούς τύπους δεδομένων. Για παράδειγμα η βιβλιοθήκη για το Surface Management διαχειρίζεται την οθόνη, η Media Framework Library χρησιμεύει για την αναπαραγωγή βίντεο και ήχων, η SQLite υποστηρίζει βάσεις δεδομένων, για τα γραφικά 2D και 3D είναι οι βιβλιοθήκες SGL και OpenGL|ES αντίστοιχα, ενώ για την πλοήγηση στο Internet υπάρχει η Webkit και η SSL για ασφαλή μεταφορά δεδομένων. Τέλος υπάρχει η βασική βιβλιοθήκη του λειτουργικού συστήματος η libc (Al Salool).

Το Android δεν περιέχει το σύστημα X Window, ούτε τις standard GNU βιβλιοθήκες, που επιβαρύνουν το σύστημα. Αντί για αυτά διαθέτει το υπόστρωμα Android Runtime (του ίδιου επιπέδου με τις υπόλοιπες βιβλιοθήκες), που περιλαμβάνει τις Java Core Libraries και το Dalvik Virtual Machine, με αποτέλεσμα να κάνει τις συσκευές Android να ξεχωρίζουν από το παραδοσιακό Linux. Οι βιβλιοθήκες πυρήνα Java χρησιμεύουν στις εφαρμογές Android που είναι γραμμένες σε Java (Wikipedia).

Το Dalvik Virtual Machine είναι μια εικονική μηχανή ανοιχτού κώδικα που είναι αναπόσπαστο μέρος του Android και τρέχει τις εφαρμογές (Wikipedia). Είναι σχεδιασμένο για να μπορεί να εκτελεί πολλές διεργασίες ταυτόχρονα, ανεξάρτητα, αποτελεσματικά και με ασφάλεια. Δηλαδή, κάθε διεργασία εκτελείται απομονωμένα σε μία εικονική μηχανή για να είναι ανεξάρτητη από τις υπόλοιπες και να μην τις επηρεάζει σε περίπτωση δυσλειτουργίας. Επίσης με τον τρόπο αυτό γίνεται καλύτερη διαχείριση μνήμης και επιτυγχάνεται μεγαλύτερη ταχύτητα εκτέλεσης. Για να δουλέψει

το Dalvik Virtual Machine μεταγλωττίζονται τα αρχεία .class που περιέχουν bytecode και παράγονται από την Java σε .dex (Dalvik Executable) (Al Salool).

Το επόμενο επίπεδο είναι το application framework το οποίο περιέχει όλα τα εργαλεία που διαχειρίζονται τις βασικές λειτουργίες του λειτουργικού και μπορεί να χρησιμοποιήσει ένας προγραμματιστής για την ανάπτυξη της εφαρμογής του. Για παράδειγμα, υπάρχουν εφαρμογές που διαχειρίζονται τα activities (στα οποία θα γίνει αναφορά παρακάτω), τα διάφορα παράθυρα, τις ειδοποιήσεις, τους πόρους του συστήματος, τις λειτουργίες του τηλεφώνου, του GPS και της κάμερας. Ουσιαστικά, είναι ένα σύνολο από εργαλεία που μπορεί να χρησιμοποιήσει ένας προγραμματιστής για να παράγει πιο σύνθετα εργαλεία (Strickland).

Τελευταίο είναι το στρώμα των εφαρμογών που περιλαμβάνει όλες τις εφαρμογές που παρουσιάζονται στους χρήστες και αλληλεπιδρούν με αυτούς. Τέτοιες εφαρμογές είναι αυτές που βρίσκονται προεγκατεστημένες στο λειτουργικό όπως οι εφαρμογές για τις επαφές, τα μηνύματα, τον browser. Επίσης υπάρχουν και οι εφαρμογές που φτιάχνονται από άλλους προγραμματιστές και οι χρήστες μπορούν να κατεβάσουν από το Android Store, όπως τα διάφορα παιχνίδια.

Τα επίπεδα της στοίβας επικοινωνούν μεταξύ τους με σκοπό να συνδέουν τα κατώτερα στρώματα του πυρήνα που διαχειρίζονται τους πόρους και το υλικό της συσκευής με τις εφαρμογές που βλέπουν και χειρίζονται οι χρήστες.

1.5 Δομή Android εφαρμογής

Υπάρχουν πολλές γλώσσες προγραμματισμού στις οποίες μπορεί να γραφτεί μία Android εφαρμογή, όπως οι C, C++, C#, και PHP. Υπάρχουν επίσης και ολόκληρες πλατφόρμες όπως το Titanium Studio που χρησιμοποιεί τις γλώσσες των web app (HTML, PHP, JavaScript, Python) και είναι συμβατή με διάφορα λειτουργικά κινητών. Ωστόσο, η Google προωθεί ιδιαίτερα την Java. Τα εργαλεία του Android SDK μεταγλωττίζουν τον κώδικα, μαζί με τους υπόλοιπους πόρους που βρίσκονται στο εκάστοτε Android package, και παράγουν ένα αρχείο με το .apk ως κατάληξη. Αυτό το αρχείο περιέχει όλον τον μεταγλωττισμένο κώδικα μιας εφαρμογής και είναι αυτό που χρησιμοποιείται στις συσκευές που έχουν Android ως εκτελέσιμο για την εγκατάσταση κάθε εφαρμογής.

Οι εφαρμογές Android αποτελούνται από τέσσερα βασικά συστατικά που καθορίζουν την συμπεριφορά τους. Καθένα από αυτά τελεί μία ξεχωριστή λειτουργία και ενώ μπορεί να εξαρτάται από άλλα συστατικά, είναι μία ξεχωριστή οντότητα. Σύμφωνα με το είδος του, κάθε συστατικό χρησιμεύει σε άλλο σκοπό και έχει διαφορετικό κύκλο ζωής που ορίζει πώς αυτό δημιουργείται και καταστρέφεται.

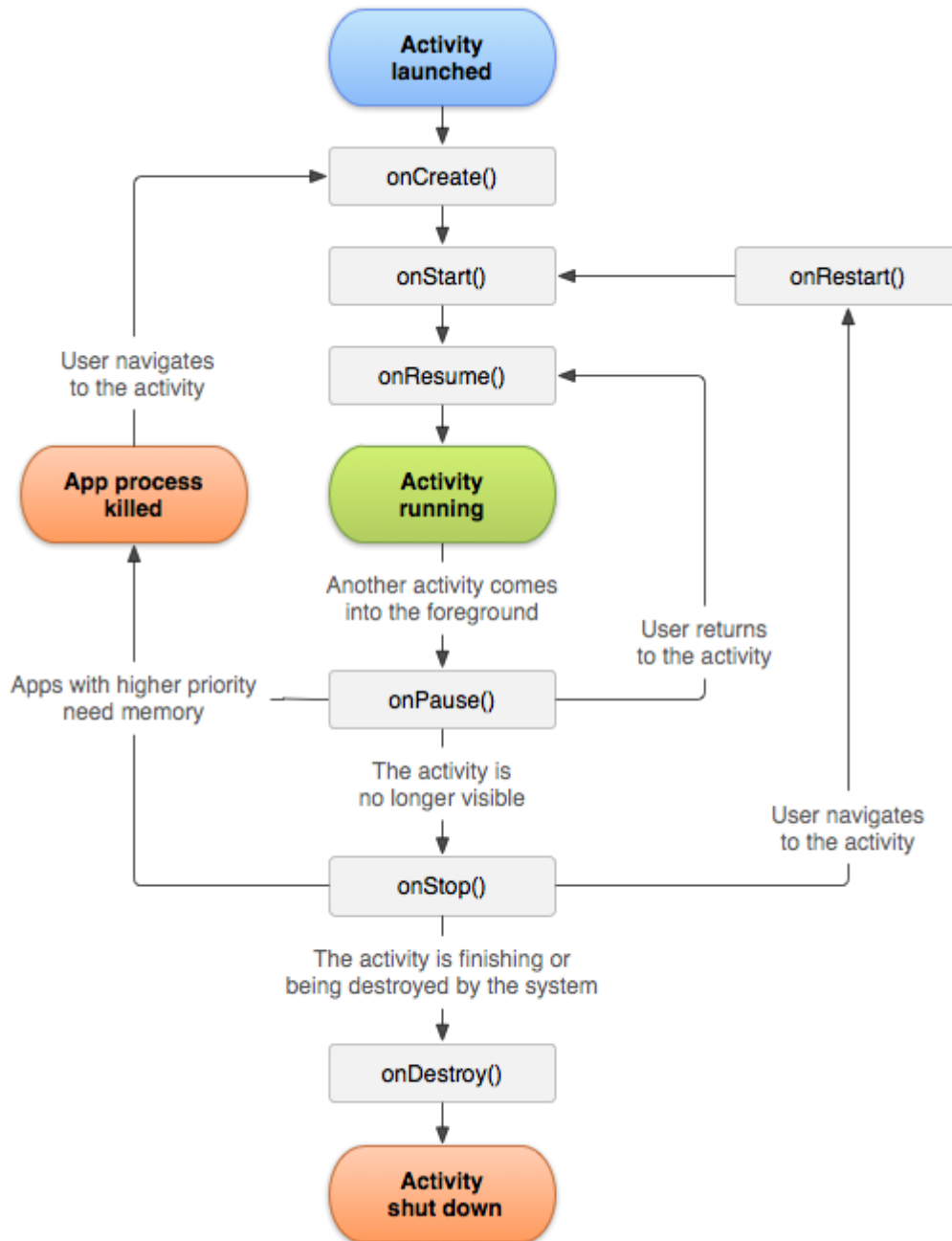
1.5.1 Activities

Σημαντικότερο συστατικό είναι τα Activities. Ένα activity αντιπροσωπεύει μία απλή οθόνη στη συσκευή με μία διεπαφή χρήστη (UI). Είναι, δηλαδή, αυτά που φαίνονται στους χρήστες και αλληλεπιδρούν με αυτούς για να κάνουν κάτι. Ένα πρόγραμμα συνήθως αποτελείται από περισσότερα του ενός activities, ανάλογα με τις διαφορετικές οθόνες που εμφανίζει, αλλά καθένα από αυτά μπορεί να υπάρχει χωρίς τα υπόλοιπα και να είναι μέρος εκκίνησης της εφαρμογής (Google Inc).

Για παράδειγμα μία εφαρμογή email μπορεί να έχει ένα activity για την προβολή εισερχόμενων, άλλο activity για τη δημιουργία νέου email, και ένα για την ανάγνωση email. Όλα τα activities συνεργάζονται για να δώσουν το καλύτερο δυνατό αποτέλεσμα στον χρήστη, μολονότι είναι ανεξάρτητα. Επίσης κάποια άλλη εφαρμογή μπορεί να ξεκινήσει κάποιο από αυτά. Λόγου χάριν, μία εφαρμογή που χειρίζεται την κάμερα μπορεί να εκκινήσει το activity που δημιουργεί νέο email, με την φωτογραφία που μόλις τραβήχτηκε ως επισυναπτόμενη (Google Inc).

Κάθε φορά που ανοίγει ένα νέο activity το προηγούμενο σταματά να βρίσκεται στο προσκήνιο, αλλά δεν διαγράφεται εκτός αν το σύστημα χρειάζεται επείγοντως πόρους. Πιο συγκεκριμένα, τρεις είναι οι κυριότερες καταστάσεις στις οποίες μπορεί να βρεθεί ένα activity: οι "resumed", "paused" και "stopped". Αρχικά, όταν το activity βρίσκεται στο προσκήνιο και ο χρήστης ασχολείται με αυτό (δεν έχει δηλαδή εμφανιστεί κάποιο άλλο activity πάνω από κάποιο κομμάτι αυτού, που να ζητά την προσοχή του) βρίσκεται στην κατάσταση "resumed" ή αλλιώς "running". Αν κάποιο άλλο activity, που δεν καλύπτει όλη την οθόνη, ή περιέχει διαφανή κομμάτια, βρεθεί πάνω από το τρέχον activity, το τρέχον activity θεωρείται "paused", μολονότι λειτουργεί ακόμα κανονικά πίσω από το νέο activity. Τρίτη περίπτωση είναι το activity να βρεθεί στην κατάσταση "stopped" όπου κάποιο νέο activity έχει καταλάβει πλήρως την οθόνη. Παρομοίως το "stopped" activity συνεχίζει να λειτουργεί κανονικά, εκτός και αν το σύστημα χρειάζεται

να ελευθερώσει πόρους. Ο κύκλος ζωής των activities που μόλις αναλύθηκε, φαίνεται και στο ακόλουθο σχήμα (Εικόνα 4) (Google Inc).



Εικόνα 4: Κύκλος ζωής ενός activity

Το activity συνεχίζει να λειτουργεί κανονικά και στις τρεις καταστάσεις, εκτός αν το σύστημα χρειαστεί πόρους. Ανάλογα με το πόσα activities τρέχουν και το πόσο έχουν "ωριμάσει" (aging) σε μία κατάσταση, είναι πιο πιθανό να σβηστούν, ξεκινώντας από αυτά που έχουν απομακρυνθεί πλήρως από το παρασκήνιο ("stopped"). Όταν όμως ο χρήστης αποφασίσει να επαναφέρει το συγκεκριμένο activity, υπάρχει πιθανότητα να

μην καταλάβει ότι είχε τερματιστεί από το σύστημα, αν ο προγραμματιστής έχει λάβει τα ανάλογα μέτρα χρησιμοποιώντας τις μεθόδους της Εικόνα 5.

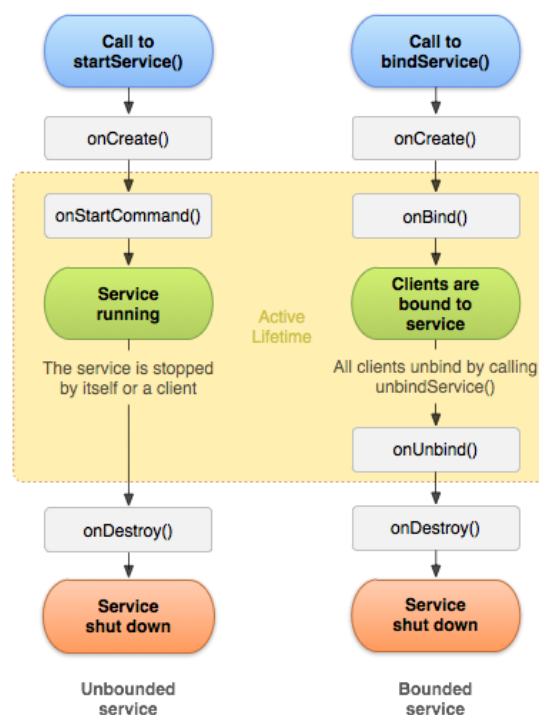
```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

Εικόνα 5: Μέθοδοι διαχείρισης καταστάσεων των activities

Επειδή τα activities παρέχουν και γραφικό περιβάλλον και επικοινωνούν άμεσα με τους χρήστες, σε περίπτωση που χρειαστεί να κάνουν κάποια διεργασία που χρειάζεται παραπάνω χρόνο από όσον ορίζει το σύστημα, ενημερώνουν τον χρήστη για την καθυστέρηση αυτή, και του δίνουν την δυνατότητα να τερματίσει την εφαρμογή. Αν κάποια εφαρμογή πρέπει να κάνει κάποια χρονοβόρα διαδικασία, είναι προτιμότερο να χρησιμοποιήσει services, που δεν εκτελούνται στο παρασκήνιο και δεν προβάλλουν διεπαφή χρήστη.

1.5.2 Services

Τα services αποτελούν τα συστατικά της εφαρμογής που εκτελούν τις χρονοβόρες διαδικασίες της εφαρμογής, αφού δεν φαίνονται στο παρασκήνιο και δεν καθυστερούν καμία άλλη διεργασία η οποία μπορεί να επικοινωνεί άμεσα με τους χρήστες. Μπορούν να συνεχίζουν να εκτελούνται ακόμα και αν ο χρήστης έχει ελαχιστοποιήσει ή κλείσει την εφαρμογή που τα κάλεσαν. Για παράδειγμα, χρησιμοποιούνται όταν κάποιο άλλο συστατικό χρειάζεται επικοινωνία με αρχεία εισόδου/εξόδου, ή διαδικτυακές συναλλαγές, ή απλά για να παίζει μουσική στο background. Τα services χωρίζονται σε δύο κατηγορίες σύμφωνα με τον τρόπο που δημιουργούνται: σε αυτά που "ξεκινάει" ("started") κάποιο άλλο συστατικό και σε αυτά που "δεσμεύονται" ("bound") από κάποιο άλλο συστατικό. Οι διαφορετικοί κύκλοι ζωής των services ανάλογα με την κατηγορία τους φαίνονται παρακάτω στην Εικόνα 6 (Google Inc).



Εικόνα 6: Κύκλος ζωής των services ανάλογα με την κατηγορία

Στην πρώτη κατηγορία, κάποιο συστατικό, όπως ένα activity, ξεκινά το service χρησιμοποιώντας την μέθοδο `startService()`. Από τη στιγμή που ξεκινήσει, το service είναι ανεξάρτητο από το συστατικό, και μπορεί να τρέχει επ' άοριστον ακόμα και αν τερματιστεί το συστατικό αυτό. Συνήθως το service εκτελεί κάποια εργασία, και δεν επιστρέφει αποτέλεσμα σε αυτόν που το δημιούργησε. Λόγου χάριν, μπορεί να

ανεβάσει ή να κατεβάσει κάποιο αρχείο από το δίκτυο και μόλις ολοκληρωθεί η εργασία αυτή να τερματιστεί από μόνο του (Google Inc).

Στην δεύτερη κατηγορία ανήκουν τα services που κάποιο συστατικό δεσμεύει με την μέθοδο `bindService()`. Αυτά τα δεσμευμένα services αλληλεπιδρούν με τα συστατικά που τα κάλεσαν στέλλοντας αιτήματα και λαμβάνοντας αποτελέσματα. Ο κύκλος ζωής τους διαρκεί όσο είναι δεσμευμένα. Υπάρχει δυνατότητα να δεσμευτούν από περισσότερα του ενός συστατικά και σε αυτήν την περίπτωση τερματίζονται όταν αποδεσμευτούν από όλα. Όπως και με τα activities, ο προγραμματιστής μπορεί να διαχειριστεί τις διάφορες καταστάσεις των services ανάλογα πάντα με την κατηγορία τους, σύμφωνα με την Εικόνα 7 (Google Inc).

```
public class ExampleService extends Service {
    int mStartMode; // indicates how to behave if the service is killed
    IBinder mBinder; // interface for clients that bind
    boolean mAllowRebind; // indicates whether onRebind should be used

    @Override
    public void onCreate() {
        // The service is being created
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // The service is starting, due to a call to startService()
        return mStartMode;
    }

    @Override
    public IBinder onBind(Intent intent) {
        // A client is binding to the service with bindService()
        return mBinder;
    }

    @Override
    public boolean onUnbind(Intent intent) {
        // All clients have unbound with unbindService()
        return mAllowRebind;
    }

    @Override
    public void onRebind(Intent intent) {
        // A client is binding to the service with bindService(),
        // after onUnbind() has already been called
    }

    @Override
    public void onDestroy() {
        // The service is no longer used and is being destroyed
    }
}
```

Εικόνα 7: Μέθοδοι για την υλοποίηση service ανάλογα με την κατηγορία τους

Παρόλο που τα services είναι ανεξάρτητα από τα activities και από τα υπόλοιπα συστατικά, τρέχουν κι αυτά εξ' ορισμού στο main thread. Για τον λόγο αυτό, όταν ένα service πρόκειται να χρησιμοποιήσει εντατικά την κεντρική μονάδα επεξεργασίας (CPU) της συσκευής πρέπει να δημιουργείται ένα νέο thread μέσα στο service. Έτσι αποφεύγονται μηνύματα προς τον χρήστη για τερματισμό της εφαρμογής λόγω καθυστέρησης (Google Inc).

1.5.3 Broadcast receivers

Το broadcast receiver είναι το συστατικό που λαμβάνει τις αναμεταδόσεις των ανακοινώσεων ολόκληρου του συστήματος, και μπορεί να διαβαστεί από τα υπόλοιπα συστατικά. Για παράδειγμα, broadcasts που παράγονται από το σύστημα είναι αυτά που ενημερώνουν για το σβήσιμο της οθόνης, για την χαμηλή μπαταρία της συσκευής, ή για μια φωτογραφία που μόλις τραβήχτηκε. Οι εφαρμογές μπορεί να παράγουν broadcasts αν θέλουν να ανακοινώσουν, λόγου χάριν, ότι ολοκληρώθηκε η λήψη ενός αρχείου και ότι είναι πλέον στην διάθεση των συνεργαζόμενων εφαρμογών (Google Inc).

Μολονότι τα broadcast receivers δεν διαθέτουν δικό τους interface, μπορούν να δημιουργούν ειδοποιήσεις στο notification bar της συσκευής. Επιπλέον, συνήθως συνδέονται με κάποιο άλλο συστατικό, όπως service, που εκτελεί κάποια εργασία βασισμένη στο γεγονός που ανακοινώθηκε (Google Inc).

Δύο είναι οι βασικές κατηγορίες broadcast που μεταδίδονται, τα normal broadcasts και τα ordered broadcasts. Τα normal broadcasts είναι ασύγχρονα και στέλνονται με την μέθοδο Context.sendBroadcast χωρίς να ενδιαφέρει η σειρά με την οποία θα μεταδοθούν στους receivers, ενώ τα ordered broadcasts που αποστέλλονται με την μέθοδο Context.sendOrderedBroadcast μεταφέρονται σε έναν receiver την φορά και ελέγχονται από προτεραιότητες (Google Inc).

Τα broadcasts receivers υλοποιούνται για να λαμβάνουν τα broadcasts, υπερφορτώνοντας την μέθοδο onReceive(Context, Intent) που περιέχει τα δεδομένα που μεταφέρονται. Μία από τις σημαντικότερες υποκλάσεις τους είναι το AppWidgetProvider που παρέχει τις απαραίτητες μεθόδους για την δημιουργία και ενημέρωση των widgets.

1.5.4 Content providers

Τα content providers είναι τα συστατικά που καθιστούν εφικτή την προσπέλαση δεδομένων μεταξύ εφαρμογών. Ουσιαστικά, διαχειρίζονται την πρόσβαση σε δομημένα σύνολα δεδομένων, δηλαδή συνήθως πίνακες. Ενθυλακώνουν τα δεδομένα και παρέχουν μεθόδους για την προσπέλασή τους με ασφάλεια. Χρησιμοποιούνται όταν μία εφαρμογή πρέπει να στέλνει και να λαμβάνει πολλά και σύνθετα δεδομένα. Για παράδειγμα, content providers υπάρχουν για τις επαφές, τα μηνύματα, τα τραγούδια που βρίσκονται στο κινητό, ακόμα και για την SQLite.

Τα δεδομένα που περιέχει ένα content provider είναι πίνακες που θυμίζουν αυτούς των σχεσιακών βάσεων δεδομένων. Για να επικοινωνήσει η εφαρμογή με το content provider ως client, αρκεί να δημιουργήσει ένα αντικείμενο τύπου ContentResolver. Με τις μεθόδους αυτού του αντικειμένου τελούνται οι βασικές ενέργειες "CRUD" (create, retrieve, update, delete). Επίσης, την γνωστή "SELECT" πραγματοποιεί η μέθοδος ContentResolver.query(), που καλεί την αντίστοιχη μέθοδο ContentProvider.query() που παρέχει η εφαρμογή-πηγή για να χρησιμοποιούνται τα δεδομένα της από άλλες εφαρμογές. Η παρακάτω Εικόνα 8 δίνει την αντιστοιχία της query() με την SELECT.

query() argument	SELECT keyword/parameter	Notes
Uri	FROM table_name	Uri maps to the table in the provider named table_name.
projection	col,col,col,...	projection is an array of columns that should be included for each row retrieved.
selection	WHERE col = value	selection specifies the criteria for selecting rows.
selectionArgs	(No exact equivalent. Selection arguments replace ? placeholders in the selection clause.)	
sortOrder	ORDER BY col,col,...	sortOrder specifies the order in which rows appear in the returned Cursor.

Εικόνα 8: Αντιστοιχία query() με SELECT

Όπως φαίνεται από την εικόνα, το Uri (Uniform resource identifier) είναι ένα String που βλέπει στον συγκεκριμένο πίνακα που πρόκειται να αναγνωστεί. Το projection αντιστοιχεί στις στήλες που η εφαρμογή ζητάει για τον πίνακα αυτό. Το selection περιέχει την συνθήκη σύμφωνα με την οποία θα επιλεγθούν οι εγγραφές του πίνακα. Το selectionArgs περιλαμβάνει τις παραμέτρους που αντιστοιχήθηκαν σε "?" κατά το selection, και τέλος το sortOrder φανερώνει τον τρόπο με τον οποίο θα οργανωθεί το επιστρεφόμενο αποτέλεσμα.

1.5.5 Intents

Μία από τις ιδιαιτερότητες του Android είναι ότι δεν υπάρχει καθορισμένο σημείο εκκίνησης στις εφαρμογές. Κάθε εφαρμογή μπορεί να ξεκινήσει από οποιοδήποτε συστατικό της, αναλόγως την περίπτωση. Αυτό είναι πολύ χρήσιμο διότι δίνει στις εφαρμογές την δυνατότητα να ξεκινάνε κομμάτια άλλων εφαρμογών, εφ' όσον διαθέτουν τις ανάλογες άδειες, όταν τα χρειάζονται. Για παράδειγμα, εάν μία εφαρμογή πρέπει να χρησιμοποιήσει την κάμερα της συσκευής, δεν είναι ανάγκη να υλοποιήσει από την αρχή ένα activity που να την χρησιμοποιεί. Μπορεί απλά να τρέξει το activity που έχει η αντίστοιχη εφαρμογή Camera και κάνει την συγκεκριμένη δουλειά (Google Inc).

Ο χρήστης έχει την εντύπωση ότι πρόκειται για την ίδια εφαρμογή και δεν καταλαβαίνει πάντα ότι οι εφαρμογές "συνεργάζονται". Στην πραγματικότητα όμως, οι εφαρμογές από μόνες τους δεν μπορούν να τρέξουν κομμάτια άλλων εφαρμογών. Μπορούν μόνο να ενημερώσουν το σύστημα για την "πρόθεσή" τους αυτή, δημιουργώντας το ανάλογο Intent. Στη συνέχεια το σύστημα αναλαμβάνει να ξεκινήσει το κατάλληλο συστατικό της εφαρμογής που ζητήθηκε, και να διεκπεραιώσει την επικοινωνία μεταξύ των εφαρμογών αυτών (Google Inc).

Τρία από τα τέσσερα συστατικά που αναλύθηκαν προηγουμένως μπορούν να ξεκινήσουν από intents: τα activities, τα services και τα broadcast receivers. Τα intents είναι ασύγχρονα μηνύματα που συνδέουν τα συστατικά μεταξύ τους κατά την εκτέλεση, ανεξάρτητα από το ποια εφαρμογή τα κάλεσε (Google Inc).

Τα intents μπορούν να είναι γενικά ή συγκεκριμένα. Όταν είναι γενικά, καλούν κάποιο activity προσδιορίζοντας κάποια ενέργεια που θέλουν να πραγματοποιηθεί, όπως η προβολή μιας φωτογραφίας ή η αποστολή ενός αρχείου. Μόλις καλέσουν το

αντίστοιχο συστατικό που χρειάζονται βάση της ενέργειας που θέλουν να πραγματοποιήσουν, το σύστημα επιστρέφει όλες τις εφαρμογές που κάνουν την ενέργεια αυτή δίνοντας στον χρήστη την επιλογή της εφαρμογής που προτιμά.

Όταν όμως τα intents είναι συγκεκριμένα, καλούνται με το URI (Uniform Resource Identifier) του συστατικού που θέλουν να εκτελεστεί. Δηλαδή ζητούν από το σύστημα να εκτελέσει συγκεκριμένο activity ή service καλώντας την κατάλληλη μέθοδο `startActivity()` ή `startService()` και περνώντας το intent με την ανάλογη ενέργεια ως παράμετρο. Για τα broadcast receivers το intent περνά ως παράμετρος στην `sendBroadcast()` και απλά περιέχει το μήνυμα που χρειάζεται να αναμεταδοθεί στην συσκευή (Google Inc).

Σε μερικές περιπτώσεις, τα intents ξεκινάνε κάποιο activity με σκοπό να επιστρέψουν κάποιο αποτέλεσμα. Το αποτέλεσμα επιστρέφεται από το activity πάλι με την μορφή intent. Για παράδειγμα αν σε μία εφαρμογή ο χρήστης πρέπει να διαλέξει κάποια από τις επαφές του κινητού για να στείλει ένα mail, καλείται ένα intent με το URI της εφαρμογής "Επαφές". Μόλις ο χρήστης επιλέξει αποστολέα επιστρέφει στην αρχική εφαρμογή με ένα intent που διαθέτει το URI του αποστολέα που επιλέχτηκε (Google Inc).

1.6 Ασφάλεια και προστασία προσωπικών δεδομένων

Το λειτουργικό σύστημα Android είναι ένα Linux σύστημα πολλαπλών χρηστών και θεωρεί κάθε εφαρμογή διαφορετικό χρήστη (διαδικασία sandbox). Για τον λόγο αυτό, αποδίδει σε κάθε εφαρμογή ένα μοναδικό Linux user ID (έναν κωδικό που μόνο το λειτουργικό γνωρίζει) και ορίζει όλες τις άδειες που έχει η εφαρμογή αυτή στους πόρους και τα αρχεία του συστήματος στο ID αυτό. Παράδειγμα αδειών είναι η χρήση της κάμερας, ή του GPS. Κάθε εφαρμογή τρέχει στη δικιά της εικονική μηχανή (VM), έτσι ώστε ο κώδικας της εφαρμογής να είναι απομονωμένος από τις υπόλοιπες εφαρμογές (Google Inc).

Με τον τρόπο αυτό, το Android εφαρμόζει την αρχή των ελάχιστων δικαιωμάτων, δίνοντας σε κάθε εφαρμογή πρόσβαση μόνο στους πόρους που χρειάζεται. Δηλαδή οι εφαρμογές δεν μπορούν να δουν μέρη του συστήματος για τα οποία δεν έχουν άδεια. Για να μοιραστεί μία εφαρμογή τα δεδομένα της με άλλες εφαρμογές ή υπηρεσίες μπορεί είτε να μοιράζεται το ίδιο Linux user ID με την άλλη εφαρμογή με αποτέλεσμα

να έχουν κοινούς πόρους και να τρέχουν στο ίδιο VM (μόνο εφ' όσον οι δύο εφαρμογές είναι υπογεγραμμένες με το ίδιο πιστοποιητικό), είτε να ζητήσει τις επιπλέον άδειες που χρειάζεται και που ο χρήστης πρέπει να παραχωρήσει κατά την εγκατάστασή της (Google Inc).

Εντούτοις, λόγω ελλιπούς τεκμηρίωσης και σύγχυσης μερικές φορές οι προγραμματιστές δηλώνουν αυθαίρετα παραπάνω δικαιώματα από όσα πραγματικά χρειάζονται οι εφαρμογές τους, με αποτέλεσμα να παραβιάζουν την αρχή των ελαχίστων δικαιωμάτων. Εταιρίες όπως οι Lookout Mobile Security, AVG Technologies, και McAfee διαθέτουν λογισμικά προστασίας από ιούς για Android, τα οποία όμως περιορίζονται και αυτά από την προαναφερθείσα διαδικασία sandbox, με αποτέλεσμα την αδυναμία ελέγχου του συστήματος σε βάθος (Wikipedia).

Οι πιο συχνές λειτουργίες που εκτελούν τα κακόβουλα λογισμικά στο Android είναι να στέλνουν μηνύματα σε νούμερα υπηρεσιών που διαθέτουν υψηλές χρεώσεις. Επίσης τέτοια λογισμικά μπορεί να αποστέλλουν προσωπικές πληροφορίες σε τρίτους, χωρίς την γνώση του χρήστη, ή να προβάλλουν ενοχλητικές διαφημίσεις. Αναφέρεται ότι τέτοιου είδους απειλές αυξάνονται εκθετικά, ωστόσο οι μηχανικοί της Google υποστηρίζουν ότι αυτές είναι υπερβολές που διαδίδουν οι εταιρίες που παράγουν λογισμικά προστασίας από ιούς, για διαφημιστικούς σκοπούς. Επίσης, η Google αναφέρει ότι μόνο το 0.5% των εφαρμογών που είχαν ανεβεί στο Google Play Store έχουν επισημανθεί ως επικίνδυνα malware (Wikipedia).

Η Google χρησιμοποιεί ένα πρόγραμμα, το Google Bouncer, για να σαρώνει τις εφαρμογές που ανεβαίνουν στο Play Store. Η λειτουργία του είναι να επισημαίνει τις ύποπτες εφαρμογές ενημερώνοντας τους χρήστες και να αφαιρεί τα κακόβουλα λογισμικά. Επίσης από την έκδοση 4.2 Jelly Bean και μετά συμπεριλαμβάνονται στο Android ένα πρόγραμμα σάρωσης των εφαρμογών που εγκαθίστανται ακόμα και από πηγές τρίτων. Επίσης προειδοποιεί τον χρήστη όταν πρόκειται να αποσταλούν μηνύματα, και δεν προχωράει στην αποστολή τους, αν δεν εγκριθεί από τον χρήστη (Wikipedia).

Οι Android συσκευές έχουν τη δυνατότητα να καταγράφουν τα στοιχεία του Access Point στο οποίο συνδέονται μέσω του Wi-Fi, καθώς κινούνται στον χώρο, δημιουργώντας βάσεις δεδομένων, από τις φυσικές τοποθεσίες εκατομμύριων AP.

Αυτές οι βάσεις σχηματίζουν ηλεκτρονικούς χάρτες που χρησιμοποιούν διάφορες εφαρμογές, όπως το Foursquare, το Google Latitude και το Facebook Places, για να εμφανίζουν διαφημίσεις σύμφωνες με την τοποθεσία της συσκευής. Τον Αύγουστο του 2013 η Google εξέδωσε το Android Device Manager που επιτρέπει στους χρήστες να εντοπίσουν, ή ακόμα και να απενεργοποιήσουν την συσκευή τους (σε περίπτωση απώλειας) μέσω μίας σελίδας στο διαδίκτυο. Αυτή η λειτουργία είναι εφικτή στις συσκευές με έκδοση 2.2 ή υψηλότερη (Wikipedia).

Το Σεπτέμβριο του 2013 έγινε γνωστό ότι Αμερικάνικες και Βρετανικές υπηρεσίες εθνικής ασφάλειας και πληροφοριών, όπως οι NSA και GCHQ, έχουν πρόσβαση στα δεδομένα των συσκευών iPhone, BlackBerry, και Android. Μπορούν δηλαδή να διαβάσουν σχεδόν όλες τις πληροφορίες των συσκευών, όπως μηνύματα, επαφές, θέσεις και emails.

1.7 Android και XML

Όπως έχει ήδη αναφερθεί, η συμβολική γλώσσα XML (Extensible Markup Language) είναι ιδιαίτερα σημαντική για την ανάπτυξη εφαρμογών σε Android. Η XML είναι ουσιαστικά μια γλώσσα που χρησιμοποιείται για την αναπαράσταση δομημένων πληροφοριών. Στην σύνταξη της, μοιάζει πάρα πολύ με την HTML (HyperText Markup Language), αφού και οι δύο γλώσσες χρησιμοποιούν την έννοια των tags, που περικλείονται από τα σύμβολα "<" και ">". Μέσα στα tags αυτά περιλαμβάνονται το περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο που παίζει το περιεχόμενο αυτό.

Στο Android, η XML χρησιμοποιείται σε τρία βασικά σημεία, που παίζουν καθοριστικό ρόλο για τις εφαρμογές: στο αρχείο manifest, στα layouts της εφαρμογής, και γενικότερα για την αναπαράσταση κάποιων τιμών (values) που χρησιμοποιούνται στον κώδικα. Τα κομμάτια που γράφονται σε XML σχετίζονται άμεσα με εκείνα που γράφονται σε Java, και αναλόγως την κατηγορία τους, εκτελούν διαφορετικές λειτουργίες.

1.7.1 AndroidManifest.xml

Το αρχείο AndroidManifest.xml, που βρίσκεται στο root directory της εκάστοτε εφαρμογής, περιλαμβάνει αναφορές για όλα τα συστατικά που χρησιμοποιεί η κάθε εφαρμογή. Είναι απαραίτητο αρχείο της εφαρμογής, επειδή με τις πληροφορίες που

περιέχει συνδέει όλα τα υπόλοιπα στοιχεία ώστε να μπορούν να δουλέψουν και να συνεργαστούν.

Μέσα στο αρχείο manifest δηλώνονται αρχικά το όνομα του package που βρίσκεται η εφαρμογή, μαζί με το νούμερο και το όνομα της έκδοσής της, όπως φαίνεται από τις παρακάτω σειρές.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="gr.aetos.programmathimatou"
    android:versionCode="1"
    android:versionName="1.0" >
```

Στη συνέχεια ορίζεται το minimum API level, και το target API level. Τα δυο αυτά πεδία δηλώνουν την ελάχιστη έκδοση Android που μπορεί να τρέξει την εφαρμογή και την έκδοση στην οποία έχει δοκιμαστεί η εφαρμογή. Τα API levels ορίζονται με ένα μοναδικό αριθμό για κάθε έκδοση, σύμφωνα με την Εικόνα 9 (Google Inc).

Platform Version	API Level	VERSION_CODE	Notes
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

Εικόνα 9: Android API levels

Όταν η εφαρμογή δημοσιευθεί στο Android Store, τότε η Google φιλτράρει τους χρήστες που μπορούν να την κατεβάσουν ανάλογα με το πώς έχουν οριστεί τα πεδία αυτά στο manifest αρχείο της εφαρμογής. Για παράδειγμα, οι παρακάτω σειρές δηλώνουν πως η συγκεκριμένη εφαρμογή δεν μπορεί να τρέξει στις εκδόσεις πριν το Froyo (2.2.x), και ότι δοκιμάζεται στην έκδοση Jelly Bean (4.2).

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
```

Ο λόγος για τον οποίο γίνεται ο διαχωρισμός αυτός είναι επειδή κάθε νέα έκδοση του Android περιέχει πιο πλούσιο API, όπως αναλύθηκε στην ιστορική αναδρομή. Επομένως κάθε εφαρμογή μπορεί να τρέξει μόνο στις συσκευές που χρησιμοποιούν το ίδιο API level ή ανώτερο.

Επίσης στο αρχείο manifest δηλώνεται αν η εφαρμογή αυτή μπορεί να γίνει backup, ποιο είναι το εικονίδιο που θα εμφανίζεται για την εκτέλεσή της και σε ποιον φάκελο βρίσκεται, ποιος είναι ο τίτλος της και πού βρίσκεται, και ποιο πρότυπο μορφοποίησης ακολουθεί, σύμφωνα με τον παρακάτω κώδικα.

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@android:style/Theme.Light.NoTitleBar" >
</application>
```

Ο συμβολισμός `"@drawable/ic_launcher"` δείχνει στον φάκελο drawable το αρχείο με όνομα ic_launcher.

Μέσα στο tag `<application>` περιλαμβάνονται όλα τα συστατικά που χρησιμοποιεί η εφαρμογή και ότι άλλο είναι αναγκαίο να γνωρίζει το σύστημα κατά την εκτέλεσή της. Τα συστατικά ορίζονται με τα εξής tags: `<activity>` για τα activities, `<service>` για τα services, `<receiver>` για τα broadcast receivers και `<provider>` για τα content providers. Απαραίτητη ιδιότητα για κάθε συστατικό είναι το όνομά του, όπως φαίνεται στον ακόλουθο κώδικα.

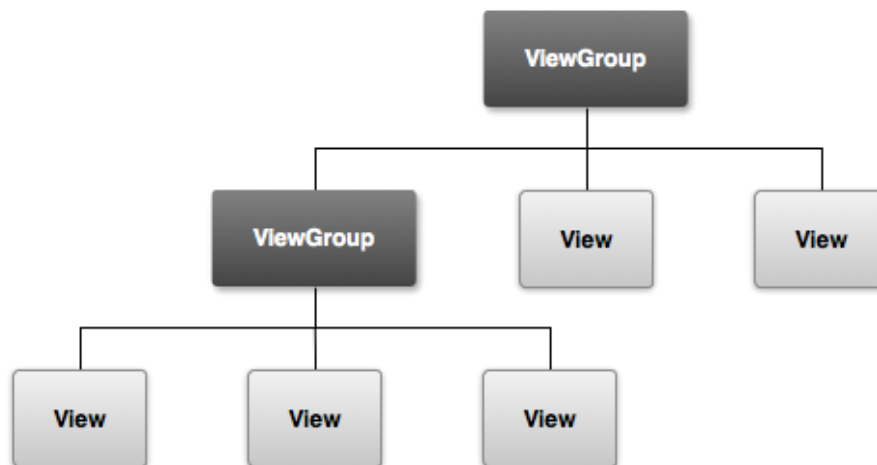
```
android:name="gr.aetos.programmamathimaton.EditModules"
```

Τέλος, το αρχείο manifest περιέχει αναφορά σε εξωτερικές βιβλιοθήκες που χρησιμοποιούνται, όπως η Google Maps library, υλικά στοιχεία ή και λογισμικό που απαιτείται να υπάρχει στις συσκευές για την ορθή λειτουργία της εκάστοτε εφαρμογής, καθώς και το σύνολο των αδειών που πρέπει να εξουσιοδοτήσει ο χρήστης στην εφαρμογή ώστε να εγκατασταθεί για να μπορεί να εκτελεστεί.

1.7.2 Layouts

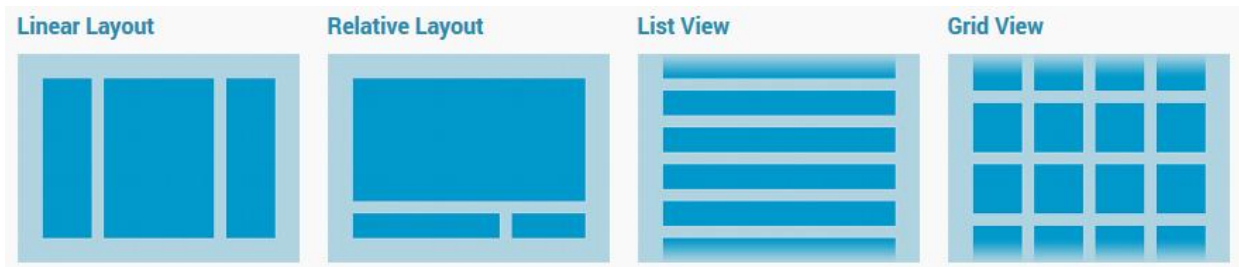
Άλλα χρήσιμα αρχεία XML των εφαρμογών που γράφονται για συσκευές Android είναι τα αρχεία που περιέχονται στον φάκελο layout που βρίσκεται μέσα στον φάκελο res (resources). Πρόκειται για τα αρχεία που ορίζουν την οπτική δομή των διεπαφών χρήστη. Θυμίζουν τον τρόπο που σχεδιάζεται το interface στην γλώσσα HTML, διότι χρησιμοποιούν και αυτά μια σειρά από εμφωλευμένα elements (Google Inc).

Κάθε διεπαφή αποτελείται από Views και ViewGroups. Το αντικείμενο View είναι αυτό που ζωγραφίζει κάτι στην οθόνη, με το οποίο ο χρήστης αλληλεπιδρά. Το ViewGroup είναι το στοιχείο που δεν φαίνεται στην οθόνη, αλλά περιέχει άλλα views και viewgroups και καθορίζει πως αυτά θα εμφανίζονται. Τα σύνολα αυτών των αντικειμένων δημιουργούν το layout που βλέπει ο χρήστης, όπως στην Εικόνα 10.



Εικόνα 10: Αναπαράσταση ενός layout από Views και ViewGroups

Τέσσερις είναι οι βασικότερες κατηγορίες των ViewGroups (Εικόνα 11). Το Linear Layout διατάσσει τα στοιχεία του οριζόντια ή κάθετα, σε σειρά. Το Relative Layout σχεδιάζει το πού βρίσκεται κάθε στοιχείο σε σχέση με τα υπόλοιπα. Το List View και το Grid View συμπληρώνονται δυναμικά μέσω των αντικειμένων AdapterView και περιέχουν τα στοιχεία τους σε μορφή λίστας ή πλέγματος, αντίστοιχα (Google Inc).



Εικόνα 11: Συνηθέστερα ViewGroups

Υπάρχουν πολλά είδη Views, καθένα από τα οποία έχει διαφορετική χρήση. Τα πιο γνωστά από αυτά είναι τα TextView, EditText, Button, CheckBox, RadioButton, ToggleButton, ProgressBar, και ImageView. Αυτά ζωγραφίζουν τα βασικότερα αντικείμενα όπως κείμενο, κουμπιά και εικόνες, που μπορεί ο χρήστης να δει και να πατήσει.

Το layout μπορεί επίσης να δημιουργηθεί δυναμικά, μέσα από τον κώδικα Java, αλλά εφόσον υπάρχουν στατικά στοιχεία, είναι προτιμότερο να ορίζονται στο XML, ώστε να διαχωρίζεται το interface από τη λειτουργικότητα της εφαρμογής.

1.7.3 Values

Μέσα στον φάκελο res της κάθε εφαρμογής υπάρχει επίσης ο φάκελος values. Εκεί τοποθετούνται διάφορα αρχεία XML που περιέχουν πληροφορίες και τιμές για διάφορα στοιχεία. Για παράδειγμα, στο color.xml ορίζονται διάφορα χρώματα που χρησιμοποιούνται στην εφαρμογή σε συγκεκριμένα σημεία, όπως στον ακόλουθο κώδικα.

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

    <color name="TITLEBAR">#999999</color>

</resources>
```

Αυτός ο κώδικας δηλώνει ότι σε οποιοδήποτε στοιχείο χρησιμοποιηθεί ως χρώμα το `"@color/TITLEBAR"` θα αντιστοιχίζεται με το δεκαεξαδικό #999999, δηλαδή μία απόχρωση του γκρι.

Στο αρχείο `dimens.xml` περιέχονται τιμές για διάφορες διαστάσεις που χρησιμοποιούνται, ενώ στο αρχείο `string.xml` αντιστοιχίζονται συμβολοσειρές σε ονόματα.

1.8 Google Play - Android Market

Από το 2008 η Google δημιούργησε μία εφαρμογή με όνομα Android Market, στην οποία οι προγραμματιστές μπορούν να ανεβάζουν τις εφαρμογές τους ώστε να είναι προσβάσιμες σε όλους τους χρήστες των Android συσκευών. Το 2012 το Android Market μετονομάζεται σε Google Play και φέρει το λογότυπο της Εικόνα 12.



Εικόνα 12: Google Play Logo

Οι εφαρμογές που περιέχονται στο Google Play είναι διαθέσιμες στους χρήστες είτε δωρεάν, είτε με κάποιο μικρό χρηματικό ποσό που κυμαίνεται από 0.50€ ως 10€, ενώ το μέγιστο όριο φτάνει τα 100€ (Google Inc). Για να τις κατεβάσουν, οι χρήστες μπορούν να συνδεθούν στο Play Store μέσα από την εφαρμογή της συσκευής τους, ή να συνδεθούν με τον ίδιο λογαριασμό Google της συσκευής τους με την ιστοσελίδα του Google Play. Όπως αναφέρθηκε και στο υποκεφάλαιο 1.7.1, ανάλογα με τις απαιτήσεις των εφαρμογών, η Google φιλτράρει τις συσκευές που δύναται να τις χρησιμοποιήσουν.

Μέχρι και το 2013 έχουν ανέβει στο Google Play πάνω από 1.000.000 εφαρμογές ενώ τα downloads τους φτάνουν τα 50.000.000.000! Για να ανεβάσει ένας προγραμματιστής μία εφαρμογή στο Google Play αρκεί να έχει developer account, το οποίο κοστίζει \$25, και να υπογράψει την εφαρμογή του ψηφιακά, επειδή οι συσκευές δεν δέχονται να εγκαταστήσουν λογισμικό που είναι υπογεγραμμένο με το debug πιστοποιητικό. Το μόνο πρόβλημα για τους προγραμματιστές που συνδέονται με λογαριασμό που δημιουργήθηκε στην Ελλάδα είναι ότι δεν έχουν δικαίωμα να ανεβάζουν εφαρμογές που κοστίζουν (Wikipedia).

1.9 Το Android Bundle IDE για ανάπτυξη εφαρμογών σε Android

Στα πλαίσια ανάπτυξης της εφαρμογής "Ατομικό Πρόγραμμα Μαθημάτων" χρησιμοποιήθηκε ως εργαλείο συγγραφής κώδικα Android, το Android Bundle. Πρόκειται για μία διανομή του γνωστού Eclipse IDE (Integrated Development Environment) το οποίο όμως περιέχει ενσωματωμένο το SDK του Android, καθιστώντας έτσι δυνατή την ανάπτυξη εφαρμογών άμεσα. Μαζί με το SDK περιλαμβάνονται και τα ADT (Android Developer Tools) που λειτουργούν και μέσω της γραμμής εντολών. Επίσης το Android Bundle διαθέτει εικονικές μηχανές προσομοιωτών συσκευών Android (Google Inc).

1.10 Επίλογος

Στο κεφάλαιο αυτό μελετήθηκαν τα χαρακτηριστικά του λειτουργικού συστήματος Android σύμφωνα με τις διάφορες εκδόσεις που κυκλοφόρησαν και περιγράφηκε η αρχιτεκτονική του και τα συστατικά που αποτελούν τις εφαρμογές. Επίσης αναφέρθηκαν τα μέτρα ασφάλειας που διαθέτει το σύστημα, η εφαρμογή Google Play που χρησιμοποιείται για την διάθεση των υπολοίπων εφαρμογών και το εργαλείο Android Bundle. Αφού πλέον υπάρχει μια πλήρης εικόνα για το λειτουργικό αυτό, το επόμενο κεφάλαιο αναλύει τη συμβολική γλώσσα αναπαράστασης αντικειμένων JSON, που θα χρησιμοποιηθεί για την εισαγωγή των δεδομένων στην εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων".

ΚΕΦΑΛΑΙΟ 2: Τεχνολογία JSON

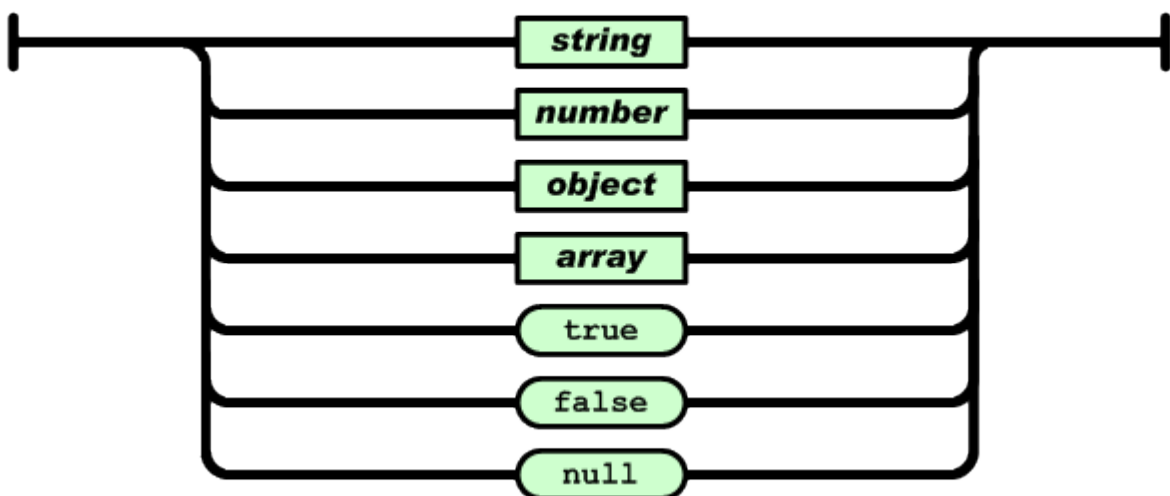
Στο κεφάλαιο εξηγείται το JSON (JavaScript Object Notation), αναλύεται η σύνταξή του, περιγράφονται οι χρήσεις του, ενώ δίνεται και ένα σύντομο παράδειγμα. Υπάρχουν αρκετοί λόγοι για τους οποίους συνιστάται η χρήση του για την ανταλλαγή δεδομένων, κυριότερος από τους οποίους είναι ότι μαθαίνεται εύκολα, και υποστηρίζεται από την πλειοψηφία των γλωσσών προγραμματισμού.

2.1 Τι είναι το JSON

Το JSON (JavaScript Object Notation) είναι μια συμβολική γλώσσα για αναπαράσταση δεδομένων. Μοιάζει με την XML (Extensible Markup Language) μόνο που είναι πιο κατανοητό από ανθρώπους και μηχανές, και συντάσσεται πιο εύκολα. Ουσιαστικά, πρόκειται για μια συμβολοσειρά (String) που όπως φανερώνει ο τίτλος του, περιγράφει αντικείμενα της γλώσσας JavaScript. Ωστόσο, λόγω της ευκολίας στην χρήση πολλές άλλες γλώσσες προγραμματισμού άρχισαν να αναπτύσσουν βιβλιοθήκες για την ανάγνωσή και γραφή του, πολλές φορές και ως συνδετικό στοιχείο των δεδομένων μεταξύ άλλων γλωσσών.

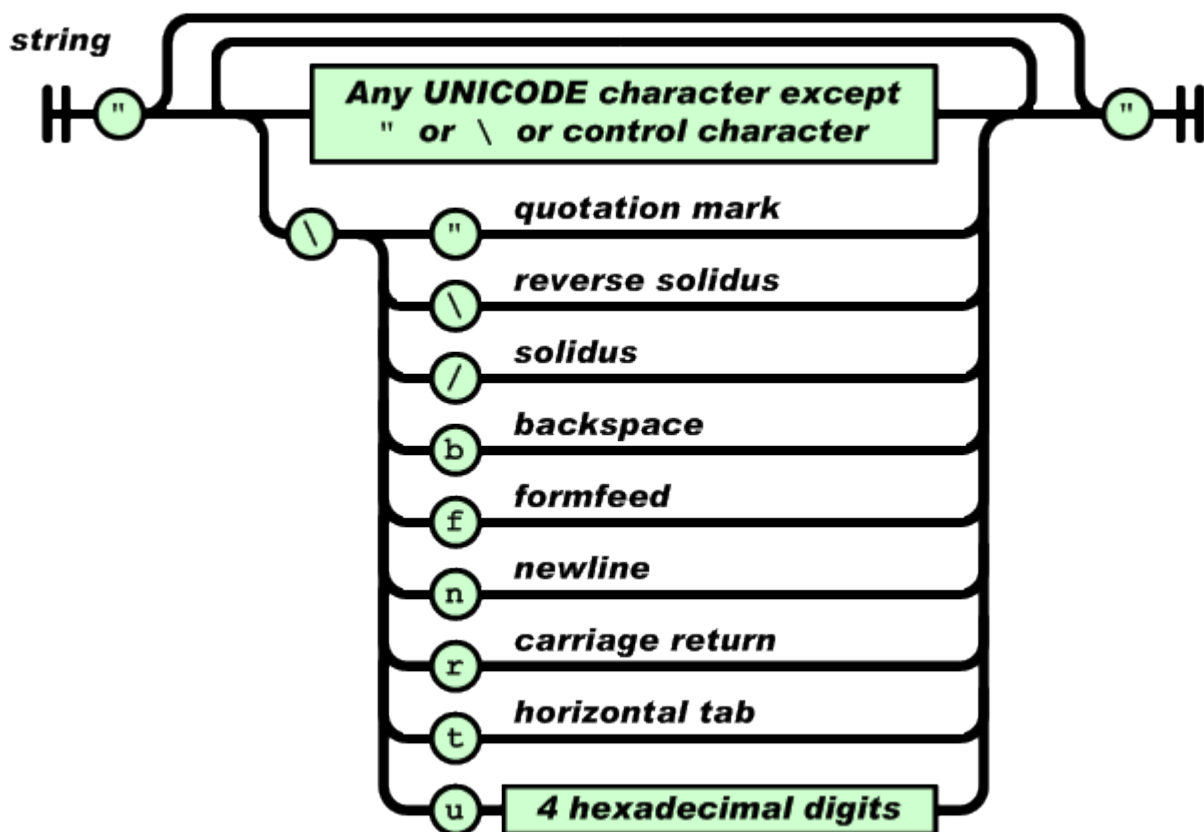
Αποτελείται από τιμές, οι οποίες μπορεί να είναι συμβολοσειρές, νούμερα, αντικείμενα, ή πίνακες, ή οι διακριτές τιμές "true", "false", "null" (Crockford), όπως φαίνεται στην Εικόνα 13:

value



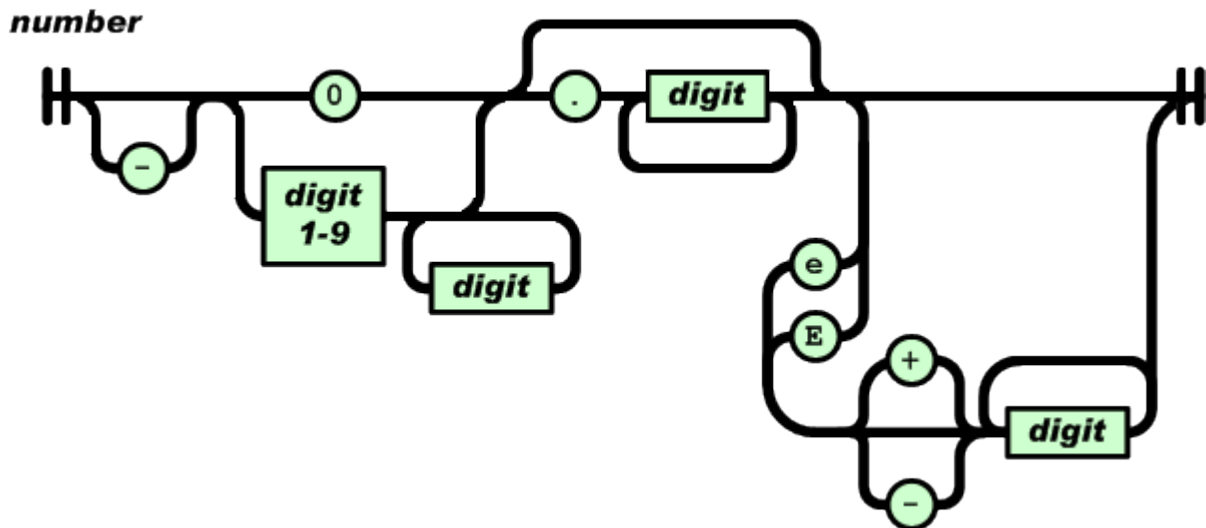
Εικόνα 13: Είδη τιμών της JSON

Κάθε String περικλείεται από εισαγωγικά (") και περιλαμβάνει οποιοσδήποτε χαρακτήρες εκτός από την κάθετο (/) και την αντίστροφη και τα εισαγωγικά ("). Υπάρχουν επίσης ειδικοί χαρακτήρες που μπορούν να αναπαρασταθούν από τον συνδυασμό της αντίστροφης καθέτου (\) με κάποιον άλλον χαρακτήρα. Αυτοί είναι οι: \ " για συμβολισμό των εισαγωγικών, \\ για την αντίστροφη κάθετο, \ / για την κάθετο, \b για το `backspace`, \f για την αλλαγή σελίδας, \n για αλλαγή γραμμής, \r για επιστροφή του δείκτη στο σημείο εκκίνησης, \t για οριζόντιο `tab`, \u μαζί με έναν τετραψήφιο δεκαεξαδικό νούμερο για αντιστοίχιση με Unicode χαρακτήρα. Παρ' όλ' αυτά, συνήθως δεν είναι ανάγκη να χρησιμοποιηθούν ειδικοί χαρακτήρες, εκτός αν χρειάζεται να εισάγουμε μεγάλο κείμενο (Crockford). Πώς παριστάνεται το String φαίνεται από το ακόλουθο σχήμα (Εικόνα 14).



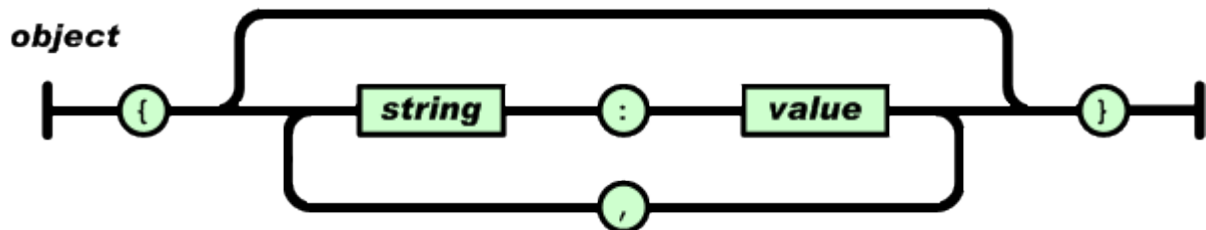
Εικόνα 14: Αναπαράσταση String σε JSON

Τα νούμερα στην JSON συμβολίζονται ακριβώς όπως συνηθίζεται γενικότερα, και αναλύεται και από το ακόλουθο διάγραμμα (Εικόνα 15):



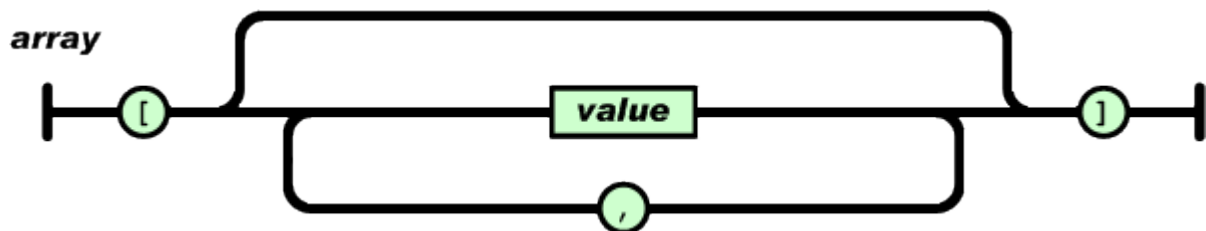
Εικόνα 15: Αναπαράσταση αριθμού σε JSON

Για να οριστεί ένα αντικείμενο με την "γλώσσα" JSON πρέπει να ξεκινά με άγκιστρο "{", να τελειώνει με άγκιστρο "}" και να περιέχει κανένα, ένα ή περισσότερα ζευγάρια της μορφής "Όνομα : Τιμή". Δίνεται όνομα στο αντικείμενο αν είναι της μορφής "Όνομα : Αντικείμενο". Στο παρακάτω σχήμα (Εικόνα 16) φαίνεται πώς γράφεται το αντικείμενο.



Εικόνα 16: Συμβολισμός αντικειμένου σε JSON

Αντίστοιχα και με τον πίνακα, για να αναπαρασταθεί σε JSON, χρειάζεται να ξεκινά με αγκύλη "[", να τελειώνει με αγκύλη "]" και να περιέχει καμία, μία ή περισσότερες τιμές, όπως δείχνει το σχήμα που ακολουθεί (Εικόνα 17).



Εικόνα 17: Συμβολισμός πίνακα σε JSON

2.2 Πώς, πότε και γιατί χρησιμοποιείται

Το JSON χρησιμοποιείται συνήθως σε περιπτώσεις που απαιτείται να εισαχθούν δεδομένα σε ένα πρόγραμμα από κάποια εξωτερική πηγή που μπορεί να μεταφέρει μόνο Strings. Επίσης, κονσόλες, όπως το Android και το Tizen έχουν τοπικούς αποθηκευτικούς χώρους για δεδομένα των εφαρμογών τους, με μοναδικό ελάττωμα ότι εκεί μπορούν να αποθηκευτούν μόνο συμβολοσειρές. Με το JSON αυτό δεν είναι πια πρόβλημα, αφού μπορεί να αναπαραστήσει αντικείμενα και πίνακες μόνο με συμβολοσειρές.

Επιπλέον, τα χαρακτηριστικά που διαθέτει για την ανταλλαγή δεδομένων είναι καταλληλότερα από αυτά της XML. Το JSON χρησιμοποιεί αποκλειστικά Unicode, κάνοντας εφικτή την επικοινωνία σε οποιαδήποτε ανθρώπινη γλώσσα. Περιγράφει εύκολα δομές και ονόματα πεδίων. Είναι πιο απλό, εύκολο, κατανοητό στη χρήση και ανάλυση, ενώ ταυτοχρόνως είναι αποτελεσματικό. Σε αντίθεση με την XML, το JSON μπορεί να αναπαραστήσει περίπλοκες δομές δεδομένων, όπως λίστες και δένδρα (Crockford).

Για τους παραπάνω λόγους, συνιστάται η χρήση του ακόμα και όταν χρειάζεται να εξαχθούν δεδομένα, από ένα πρόγραμμα που είναι γραμμένο σε μία γλώσσα προγραμματισμού, τα οποία θα είναι το input προγράμματος σε άλλη γλώσσα. Επίσης σε αρκετές περιπτώσεις χρησιμοποιείται και όταν πρόκειται να υπάρχουν ρυθμίσεις για αλλαγή γλώσσας σε ένα πρόγραμμα, περιέχοντας την ίδια δομή δεδομένων ξεχωριστά για κάθε γλώσσα.

2.3 Παράδειγμα

Έστω ότι συναντάται ο παρακάτω κώδικας.

```
Mathima = {
  id: 4101,
  onoma: "Εισαγωγή στη Πληροφορική",
  eksamino: "A",
  didaskontes: ["Ηλιούδης Χρήστος", "Γιακουστίδης Κωσταντίνος", "Βίτσας Βασίλειος", "Αντωνίου Στάθης"]
}
```

Αυτός ο κώδικας εννοεί πως υπάρχει ένα αντικείμενο τύπου μάθημα, το οποίο περιέχει τα πεδία id, onoma, eksamino, didaskontes. Το πρώτο από αυτά είναι integer, τα δύο επόμενα String, ενώ το τελευταίο περιέχει έναν πίνακα από String.

2.4 Επίλογος

Είδαμε τις χρήσεις του JSON, την σύνταξή του καθώς και ένα σύντομο παράδειγμα. Αφού έχει γίνει και η ανάλογη αναφορά στην πλατφόρμα Android στο προηγούμενο κεφάλαιο, έχουν πλέον αποσαφηνιστεί οι βασικότερες τεχνολογίες που θα χρειαστούν για την ανάλυση και ανάπτυξη της εφαρμογής "Προγράμματα Μαθημάτων", που περιγράφεται στο επόμενο κεφάλαιο.

ΚΕΦΑΛΑΙΟ 3: Η Εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων"

Στο κεφάλαιο αυτό αναλύεται η εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων" για το τμήμα Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Αρχικά αναφέρονται οι στόχοι από τους οποίους προκύπτουν οι απαιτήσεις της εφαρμογής. Στη συνέχεια περιγράφεται η πηγή δεδομένων από την οποία τροφοδοτείται η εφαρμογή με το συνολικό πρόγραμμα του τμήματος. Ακολουθεί το διάγραμμα κλάσεων με επεξήγηση της κάθε οντότητας, ο σχεδιασμός κάθε διεπαφής, καθώς και πλήρης ανάλυση της λειτουργικότητας του προγράμματος βήμα προς βήμα. Επιπλέον επεξηγούνται οι χρήσεις των AsyncTask και Serializable Interface για την ορθότερη και ομαλότερη λειτουργία της εφαρμογής. Τέλος, περιγράφεται η δημιουργία του widget και η χρήση του service για την ενημέρωση του widget αυτού, που έγιναν στοχεύοντας στην διευκόλυνση και άμεση ενημέρωση του χρήστη.

3.1 Ορισμός απαιτήσεων

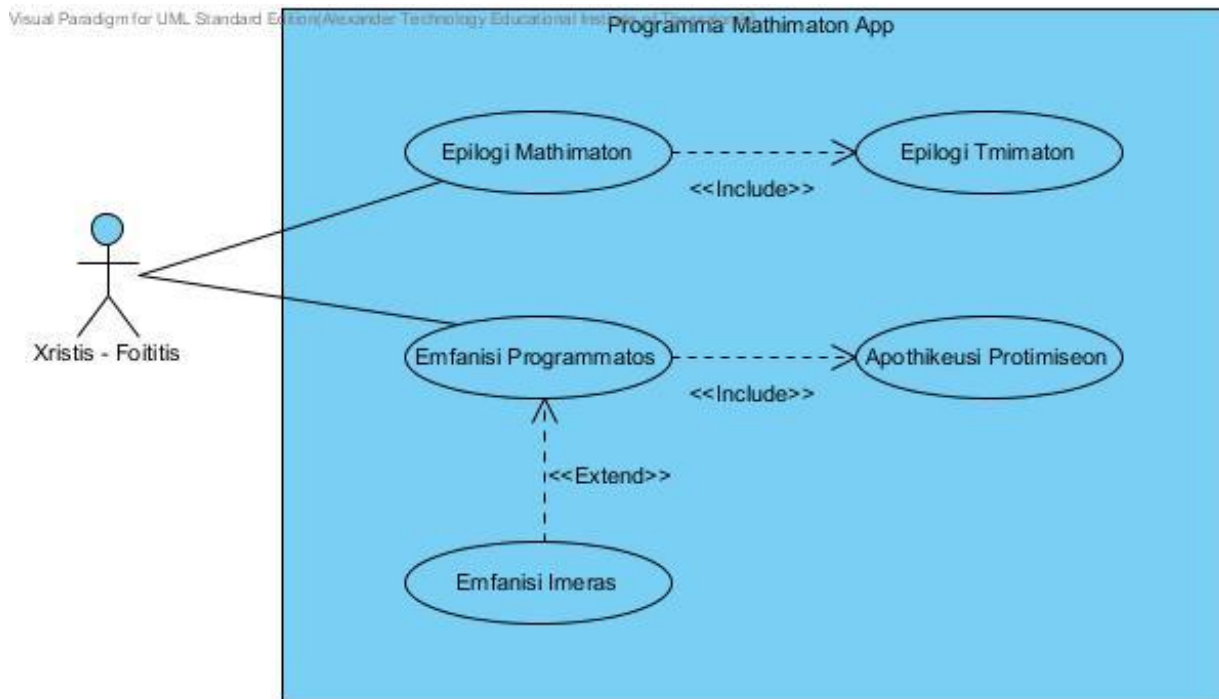
Ο κυριότερος στόχος της εφαρμογής είναι να μπορεί ο χρήστης-φοιτητής να δει το ατομικό του πρόγραμμα μαθημάτων για το τρέχον εξάμηνο. Για να γίνει αυτό πρέπει πρώτα να επιλέξει ποια μαθήματα θέλει να παρακολουθήσει από την λίστα μαθημάτων. Αφού επιλέξει το μάθημα, πρέπει να διευκρινίσει τα θεωρητικά ή εργαστηριακά τμήματα του μαθήματος που θέλει να εμφανίζονται. Για να ικανοποιηθούν οι στόχοι της εφαρμογής, πρέπει να οριστούν σαφώς οι λειτουργίες που θα υλοποιούν τις απαιτήσεις που θέτονται.

Αρχικά η εφαρμογή δίνει στον φοιτητή την δυνατότητα επιλογής των μαθημάτων που θέλει να παρακολουθήσει και μόλις ο χρήστης-φοιτητής πατήσει σε κάποιο από αυτά η εφαρμογή πρέπει να ρωτάει τον φοιτητή ποια από τα τμήματα (θεωρητικά και εργαστηριακά) τον ενδιαφέρει να εμφανίζονται. Δεν θεωρήθηκε απαραίτητο να υπάρχει περιορισμός των τμημάτων, επειδή υπάρχει η πιθανότητα ο φοιτητής να έχει δηλώσει άλλο τμήμα και να παρακολουθεί άλλο, αλλά να θέλει να εμφανίζονται και τα δύο στο πρόγραμμά του.

Στη συνέχεια ο φοιτητής επιλέγει την αποθήκευση των μαθημάτων του πατώντας το αντίστοιχο εικονικό κουμπί, που τον μεταφέρει στην οθόνη προβολής του ατομικού του προγράμματος. Εκεί μπορεί να δει ποια μαθήματα από αυτά που έχει επιλέξει πραγματοποιούνται σε ποιες μέρες και ώρες, ενώ πατώντας σε μια συγκεκριμένη

μέρα, βλέπει τα μαθήματα που έχει για την μέρα αυτή. Μπορεί επίσης να τροποποιήσει το πρόγραμμά του πατώντας το κουμπί που τον μεταφέρει στην λειτουργία "Επιλογή Μαθημάτων".

Οι αλληλεπιδράσεις αυτές του χρήστη και της εφαρμογής φαίνονται στο ακόλουθο διάγραμμα χρηστών - στόχων (Εικόνα 18).



Εικόνα 18: Διάγραμμα χρηστών - στόχων

Όπως γίνεται αντιληπτό, ο φοιτητής έχει τις εξής δυνατότητες: επιλογή μαθημάτων και εμφάνιση προγράμματος. Για να ολοκληρωθεί η επιλογή μαθημάτων πρέπει να σημειωθούν τα τμήματα που επιθυμεί να καταγράφονται, ενώ για να εμφανιστεί το πρόγραμμα πρέπει να έχουν σωθεί οι προτιμήσεις του, και αφού γίνει αυτό μπορεί να διαλέξει κάποια συγκεκριμένη μέρα για προβολή των μαθημάτων που πραγματοποιούνται την ημέρα εκείνη.

3.2 Πηγή δεδομένων

Όπως είναι λογικό, κάπως πρέπει η εφαρμογή να γνωρίζει το πρόγραμμα μαθημάτων του τμήματος ώστε να μπορεί στην συνέχεια να επεξεργαστεί την πληροφορία αυτήν και να την εμφανίσει κατάλληλα. Για τον λόγο αυτό μέσα στον φάκελο raws που

περιέχεται μέσα στα αρχεία του προγράμματος βρίσκεται ένα αρχείο μορφής JSON που περιλαμβάνει τα δεδομένα (Παράρτημα 1).

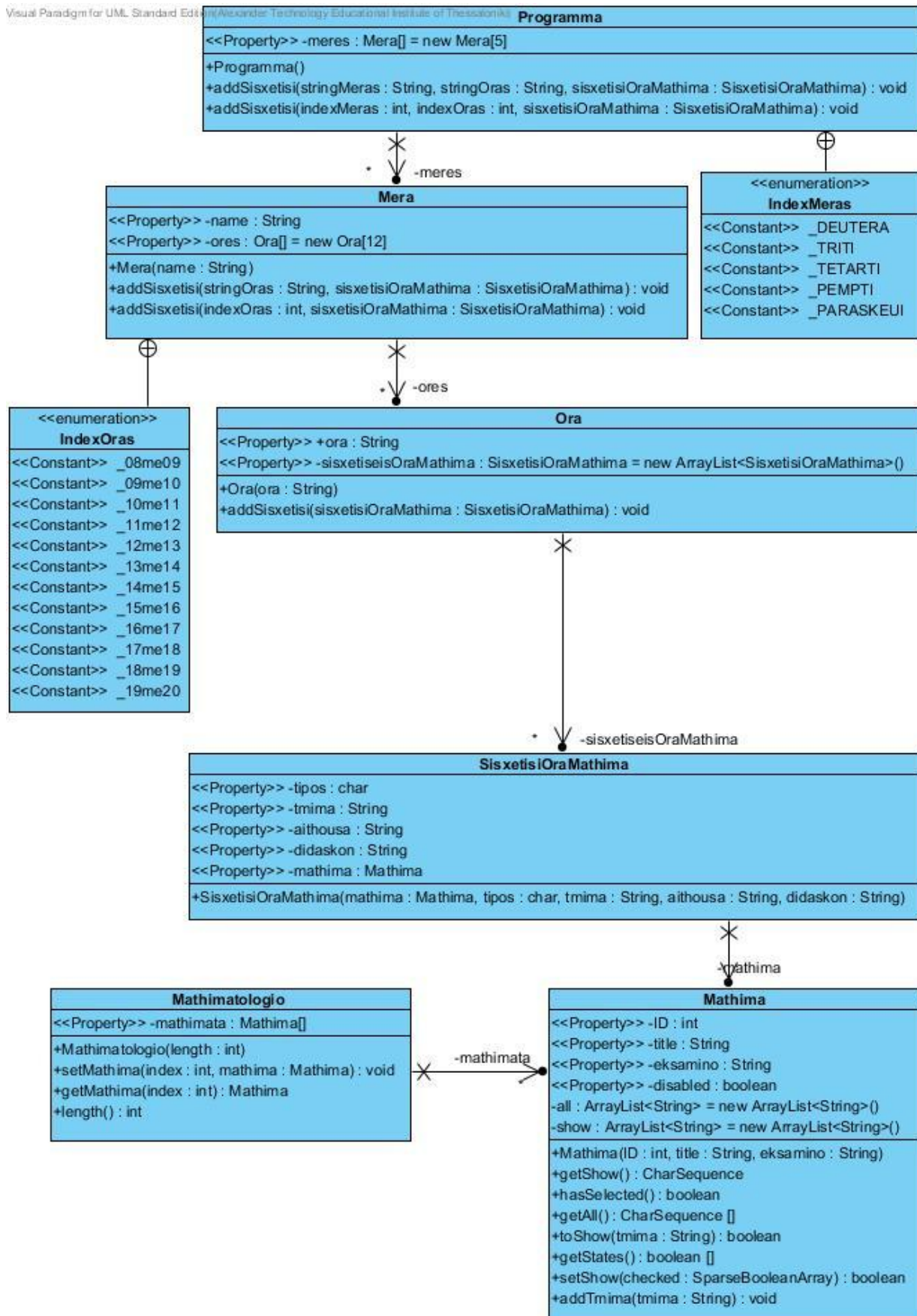
Το αρχείο αυτό αναπαριστά τα μαθήματα ως αντικείμενα σε έναν πίνακα. Κάθε αντικείμενο μάθημα διαθέτει τα πεδία id, title, eksamino, sisxetiseisOraMathima, τα οποία είναι με τη σειρά ο κωδικός του μαθήματος, το όνομά του, το εξάμηνο στο οποίο το συγκεκριμένο μάθημα διδάσκεται, και ένας πίνακας που περιέχει αντικείμενα που σχετίζονται με το πότε πραγματοποιείται κάθε μάθημα. Αυτές οι συσχετίσεις προκύπτουν επειδή η συσχέτιση ώρα-μάθημα είναι πολλά προς πολλά και για τον λόγο αυτό έχουν τις εξής ιδιότητες mera, ora, tipos, tmima, aithousa και didaskon, που αντιπροσωπεύουν την μέρα και ώρα που πραγματοποιείται η συγκεκριμένη διάλεξη, τον τύπο της (θεωρία ή εργαστήριο), το θεωρητικό ή εργαστηριακό τμήμα, την αίθουσα στην οποία λαμβάνει χώρα και τον διδάσκοντα καθηγητή της αντίστοιχα.

Εδώ πρέπει να σημειωθεί ότι ενώ το JSON αποθηκεύει τα πάντα ως συμβολοσειρές, έχει κωδικοποιηθεί έτσι ώστε να γράφεται πιο εύκολα και ταυτοχρόνως να μπορεί να χρησιμοποιηθεί σωστά και από τις κλάσεις που θα περιέχουν τα δεδομένα. Δηλαδή, οι μέρες περιέχουν μόνο τις τιμές "_DEUTERA", "_TRITI", "_TETARTI", "_PEMPTI", "_PARASKEUI" και οι ώρες μόνο τις τιμές "_09me10", "_10me11", "_11me12", "_12me13", "_13me14", "_14me15", "_15me16", "_16me17", "_17me18", "_18me19", "_19me20". Ο τύπος είναι το γράμμα "θ", ή "T", η αίθουσα τα γράμματα "ΑΜΦ", ή κάποιο τριψήφιο νούμερο, και ο διδάσκων δύο λέξεις. Αυτές οι συμβολοσειρές στη συνέχεια θα διαβαστούν αναλόγως από τις κλάσεις που θα δημιουργηθούν, και θα αρχικοποιηθούν με τους σωστούς τύπους δεδομένων.

Η εφαρμογή, κατά την πρώτη εκτέλεση της διαβάζει τα στοιχεία αυτά και στη συνέχεια δημιουργεί αντικείμενα της Java, τα οποία έχουν παρόμοια πεδία, και θα αποθηκεύονται κάθε φορά που ο χρήστης σώσει τις προτιμήσεις του. Το πώς γίνεται αυτή η διαδικασία θα ακολουθήσει στα επόμενα υποκεφάλαια.

3.3 Διάγραμμα κλάσεων

Οι κύριες κλάσεις που δημιουργούνται και χρησιμοποιούνται για την επεξεργασία και αποθήκευση των δεδομένων της εργασίας φαίνεται στο ακόλουθο διάγραμμα (Εικόνα 19).



Εικόνα 19: Διάγραμμα κλάσεων για τα δεδομένα

Αρχικά βλέπουμε το πρόγραμμα. Περιέχει έναν πίνακα πέντε αντικειμένων ημέρας, καθένα από τα οποία με την σειρά του περιέχει έναν πίνακα δώδεκα αντικειμένων ώρας. Το πρόγραμμα επίσης διαθέτει και ένα enumeration, το IndexMeras, ενώ η μέρα ένα άλλο, το IndexOras. Τα δύο αυτά enumerations βοηθάνε την προσπέλαση των στοιχείων των δύο πινάκων με βάση κάποιο String, το οποίο αντιστοιχίζεται στο JSON, για παράδειγμα:

```
int indexMeras = IndexMeras.valueOf("_TRITI").ordinal();
```

```
int indexOras = IndexOras.valueOf("_15me16").ordinal();
```

```
Ora ora = meres[indexMeras].getOres()[indexOras];
```

Το οποίο θα μας επέστρεφε το αντικείμενο Ora που περιέχει μια λίστα με συσχετίσεις μαθημάτων για την ημέρα Τρίτη, στις 15:00-16:00. Έτσι είναι πιο εύκολο να βρούμε μια μέρα και ώρα με βάση το όνομά της, και όχι σύμφωνα με τον δείκτη της στον πίνακα που τις περιλαμβάνει.

Η λίστα με τις συσχετίσεις αντιστοιχεί στις συσχετίσεις του JSON δηλαδή για συγκεκριμένη μέρα και ώρα κάθε συσχέτιση της λίστας περιέχει τα στοιχεία char tipos, String tmima, String aithousa, String didaskon, Mathima mathima, τα οποία πλέον είναι ορισμένα όπως αποφασίστηκε. Το αντικείμενο μάθημα με την σειρά του περιέχει τα χαρακτηριστικά του μαθήματος int ID, String title, String eksamino, boolean disabled, τα οποία δείχνουν τον μοναδικό κωδικό του μαθήματος, τον τίτλο του, το εξάμηνο στο οποίο ανήκει, και το αν υπάρχουν διαλέξεις για το τρέχον εξάμηνο ή όχι.

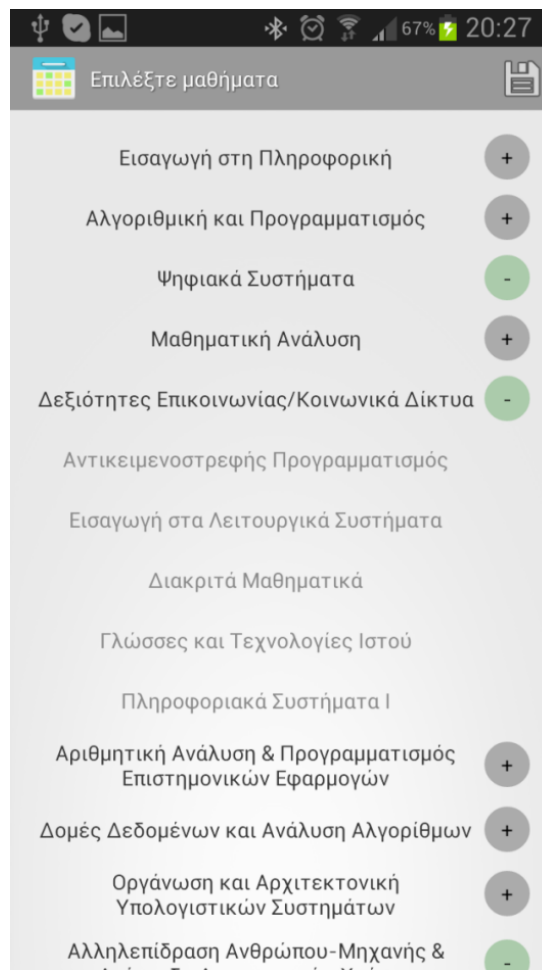
Στο αντικείμενο mathima περιέχονται επίσης και δυο λίστες, η all και η show, η πρώτη από τις οποίες περιέχει όλα τα τμήματα του συγκεκριμένου μαθήματος και η δεύτερη περιέχει τα τμήματα που ο φοιτητής-χρήστης έχει αποφασίσει να εμφανίζονται στο πρόγραμμα της εφαρμογής του.

Αντικείμενα τύπου Mathima περιλαμβάνει και το αντικείμενο Mathimatologio. Το Mathimatologio έχει ως ιδιότητα έναν πίνακα που περιέχει όλα τα μαθήματα ανεξάρτητα αν διδάσκονται το συγκεκριμένο εξάμηνο ή όχι. Αυτό μας είναι χρήσιμο για άμεση προσπέλαση των μαθημάτων, χωρίς να χρειάζεται να διαβαστεί ολόκληρο το πρόγραμμα. Άλλωστε οι δύο πίνακες αυτοί (πρόγραμμα και μαθήματα) περιέχουν απλά μία αναφορά στο κάθε μάθημα, και όχι ολόκληρο το στιγμιότυπο.

3.4 Σχεδίαση της εφαρμογής

Θέλουμε η εφαρμογή να γίνει όσο το δυνατόν απλούστερη και εύχρηστη. Για τον λόγο αυτό θα προτιμηθούν απλές και περιεκτικές οθόνες, που θα περιλαμβάνουν όλες τις απαραίτητες πληροφορίες και λειτουργίες. Επίσης εφ' όσον ο χρήστης έχει ενεργοποιημένο το rotation της συσκευής μπορεί ανά πάσα στιγμή να την γυρίσει πλάγια για να δει τις οθόνες αυτές σε landscape μορφή, όπου τα δεδομένα φαίνονται πιο συνολικά. Δύο είναι λοιπόν οι κυριότερες οθόνες που θα περιέχει η εφαρμογή: η επιλογή μαθημάτων και η προβολή του προγράμματος.

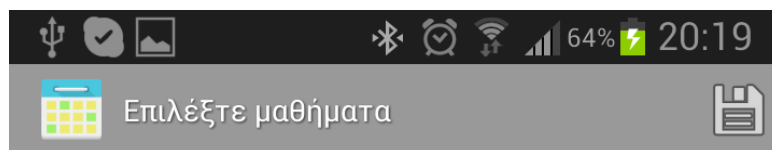
Στην επιλογή μαθημάτων εμφανίζεται μια λίστα με το σύνολο των μαθημάτων του τμήματος. Περιέχονται δηλαδή και τα μαθήματα που δεν πραγματοποιούνται αυτό το εξάμηνο με γκρι γράμματα. Κάθε σειρά της λίστας που περιέχει ένα μάθημα που πραγματοποιείται αυτό το εξάμηνο έχει και ένα κουμπί με ένα συν "+" για την επιλογή του μαθήματος, όπως φαίνεται στην Εικόνα 20.



Εικόνα 20: Εικόνα επιλογής μαθημάτων

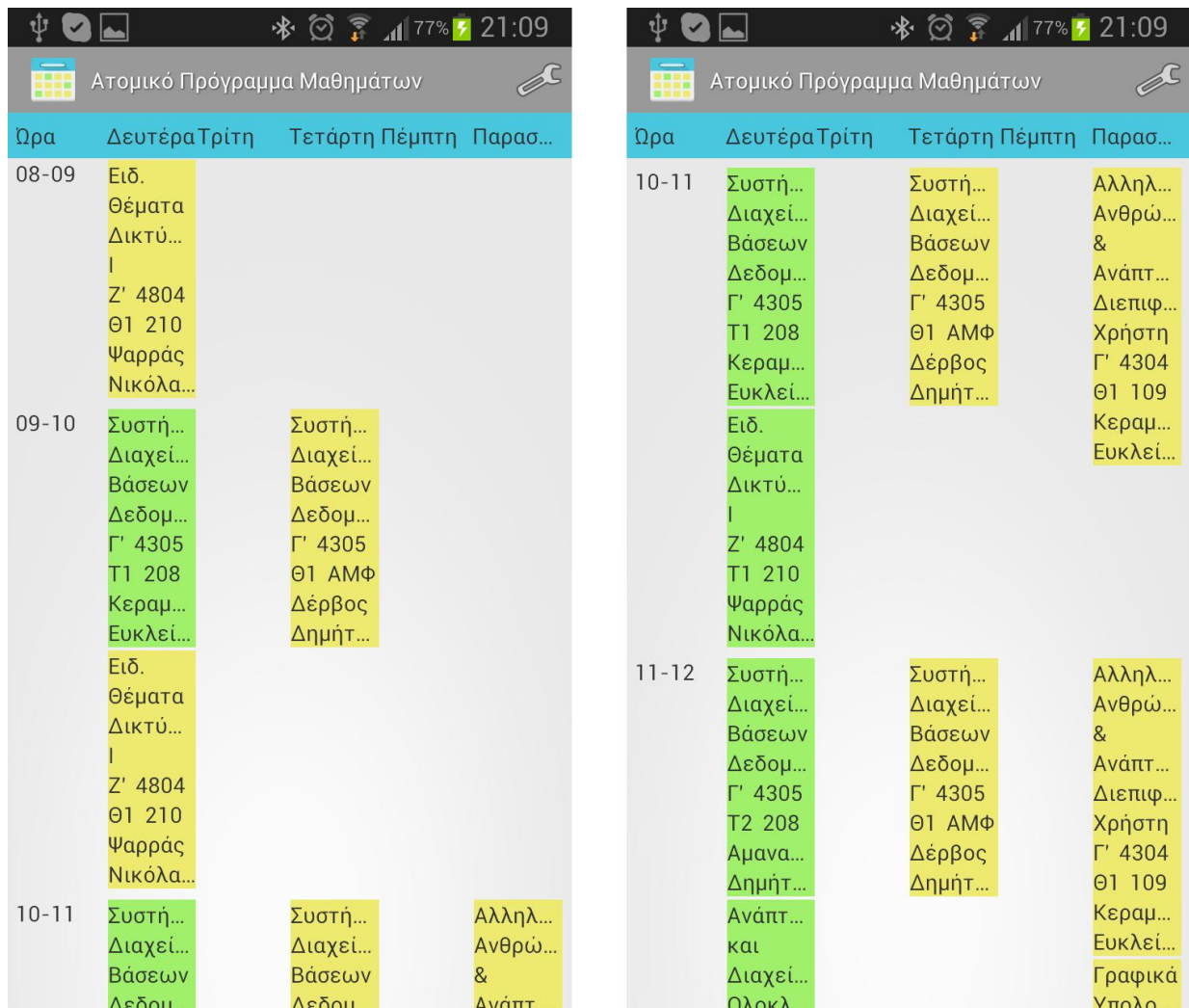
Όταν ο χρήστης έχει επιλέξει έστω και ένα από τα τμήματα κάποιου μαθήματος, το κουμπί δίπλα από το μάθημα αυτό γίνεται πράσινο με τον χαρακτήρα μείον "-".

Ακριβώς πάνω από τη λίστα των μαθημάτων υπάρχει η γραμμή ενεργειών της εφαρμογής (action bar) που περιέχει το εικονίδιο της εφαρμογής, ένα σχόλιο για την τρέχουσα οθόνη, και ένα κουμπί. Η γραμμή αυτή δημιουργήθηκε με κώδικα (Παράρτημα 2), αντί να χρησιμοποιηθεί το API Level 11 της έκδοσης 3.0, που εισάγει τα action bars, διότι αν γινόταν αυτό η εφαρμογή θα περιοριζόταν μόνο στις συσκευές που τρέχουν την έκδοση αυτή και άνω. Στην περίπτωση του activity EditModules το κουμπί έχει το εικονίδιο της αποθήκευσης όπως φαίνεται στην ακόλουθη Εικόνα 21.



Εικόνα 21: Action bar για την επιλογή μαθημάτων

Μόλις πατηθεί το κουμπί της αποθήκευσης, εμφανίζεται η δεύτερη οθόνη που περιέχει το ατομικό πρόγραμμα μαθημάτων. Αυτή η οθόνη περιέχει παρόμοιο action bar με διαφορετικό τίτλο και διαφορετικό κουμπί. Το κουμπί πλέον δείχνει ένα εργαλείο, που φανερώνει ότι από εκεί ο χρήστης μπορεί να τροποποιήσει τα μαθήματα που επέλεξε να φαίνονται, δηλαδή να περάσει στην προηγούμενη οθόνη. Κάτω από τη γραμμή ενεργειών υπάρχει ο πίνακας που περιέχει τις μέρες και ώρες που πραγματοποιούνται τα μαθήματα που έχει επιλέξει ο χρήστης, με τις θεωρίες να έχουν μια κίτρινη απόχρωση, και τα εργαστήρια μια πράσινη για να ξεχωρίζουν, όπως φαίνεται στην παρακάτω Εικόνα 22.



Εικόνα 22: Οθόνες προβολής ατομικού προγράμματος μαθημάτων

Μόλις ο χρήστης γύρει το κινητό στο πλάι και αν έχει ενεργοποιημένη την κατάλληλη ρύθμιση, η οθόνη αυτή γυρνάει σε landscape, για να φαίνονται ολόκληρες οι λέξεις. Αλλαγές στην landscape προβολή μπορούν να γίνουν και μέσω του layout απλά δημιουργώντας έναν νέο φάκελο μέσα στον φάκελο "res" που να λέγεται "layout-land". Εκεί τοποθετείται το XML που διαμορφώνει το layout για το landscape, διατηρώντας το ίδιο όνομα με το αντίστοιχο του portrait. Στο landscape φαίνεται και το πεδίο "τύπος" του μαθήματος ('Θ' ή 'Ε') όπως φαίνεται στις Εικόνα 23 και Εικόνα 24.

Ωρα	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
11-12	Συστήματα Διαχείρισης Βάσεων Δεδομένων Γ' 4305 Ε Τ2 208 Αμανατιάδης Δημήτριος Ανάπτυξη και Διαχείριση Ολοκληρωμέν... Πλ.	Ανάπτυξη και Διαχείριση Ολοκληρωμέν... Πλ. Συστημάτων & Εφαρμογών Ζ' 4701 Ε Τ2 301 Σαλαμπάσης Μιχαήλ	Συστήματα Διαχείρισης Βάσεων Δεδομένων Γ' 4305 Θ Θ1 ΑΜΦ Δέρβος Δημήτριος		Αλληλεπίδραση Ανθρώπου-Μ... & Ανάπτυξη Διεπιφανειών Χρήστη Γ' 4304 Θ Θ1 109 Κεραμόπουλος Ευκλείδης Γραφικά Υπολογιστών Ζ' 4808

Εικόνα 23: Οθόνη προβολής ατομικού προγράμματος μαθημάτων σε landscape 1

Ωρα	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
10-11	Συστήματα Διαχείρισης Βάσεων Δεδομένων Γ' 4305 Ε Τ1 208 Κεραμόπουλος Ευκλείδης Ειδ. Θέματα Δικτύων Ι Ζ' 4804		Συστήματα Διαχείρισης Βάσεων Δεδομένων Γ' 4305 Θ Θ1 ΑΜΦ Δέρβος Δημήτριος		Αλληλεπίδραση Ανθρώπου-Μ... & Ανάπτυξη Διεπιφανειών Χρήστη Γ' 4304 Θ Θ1 109 Κεραμόπουλος Ευκλείδης

Εικόνα 24: Οθόνη προβολής ατομικού προγράμματος μαθημάτων σε landscape 2

3.5 Λειτουργικότητα προγράμματος

Αρχικά προτιμάται μόνο για την πρώτη εκτέλεση του προγράμματος να εμφανίζεται το activity που υλοποιεί την επιλογή μαθημάτων. Αυτό γίνεται επειδή σίγουρα κατά την πρώτη εκτέλεση δεν υπάρχουν επιλεγμένα μαθήματα για προβολή, οπότε δεν έχει νόημα να εμφανίζεται ένα κενό πρόγραμμα. Για να καταλάβει η εφαρμογή ότι είναι η πρώτη εκτέλεση, απλά ελέγχει αν υπάρχει αποθηκευμένο κάποιο από τα δεδομένα που σώζει χρησιμοποιώντας τον παρακάτω κώδικα.

```
// Get intent's extras
Bundle extras = getIntent().getExtras();

// check if extras are empty
if(extras == null) {
```

```

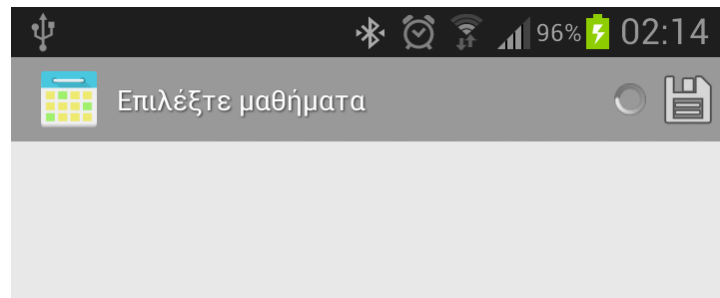
// if extras are empty, read memory
ProgrammaBundle programmaBundle =
(ProgrammaBundle)Cacher.readObjectFromFile(this, "programmaBundle");
// check if memory is empty
if(programmaBundle == null){
    // if memory is empty, start activity EditModules
    startEditModulesActivity();
} else {
    // if memory is NOT empty, fill Views
    programma = (Programma) programmaBundle.getProgramma();
    mathimatologio =
(Mathimatologio)programmaBundle.getMathimatologio();
    fillViews(programma.getMeres());
}
} else {
    // if extras are NOT empty, fill Views
    ProgrammaBundle programmaBundle =
(ProgrammaBundle)extras.getSerializable("programmaBundle");
    programma = (Programma) programmaBundle.getProgramma();
    mathimatologio = (Mathimatologio)programmaBundle.getMathimatologio();
    fillViews(programma.getMeres());
}
}

```

Ο κώδικας αυτός αρχικά βλέπει αν το intent που δημιούργησε το activity έχει extras τύπου Bundle, δηλαδή μεταφέρει δεδομένα, και αν όχι, τότε κοιτάει αν υπάρχει κάτι αποθηκευμένο στην μνήμη του κινητού. Αν δεν υπάρχει τίποτα αποθηκευμένο για την εφαρμογή αυτή, τότε είναι η πρώτη της εκτέλεση, και άρα ξεκινά το EditModules activity με την μέθοδο startEditModulesActivity(), αλλιώς εμφανίζει το πρόγραμμα μαθημάτων με την μέθοδο fillViews(programma.getMeres());.

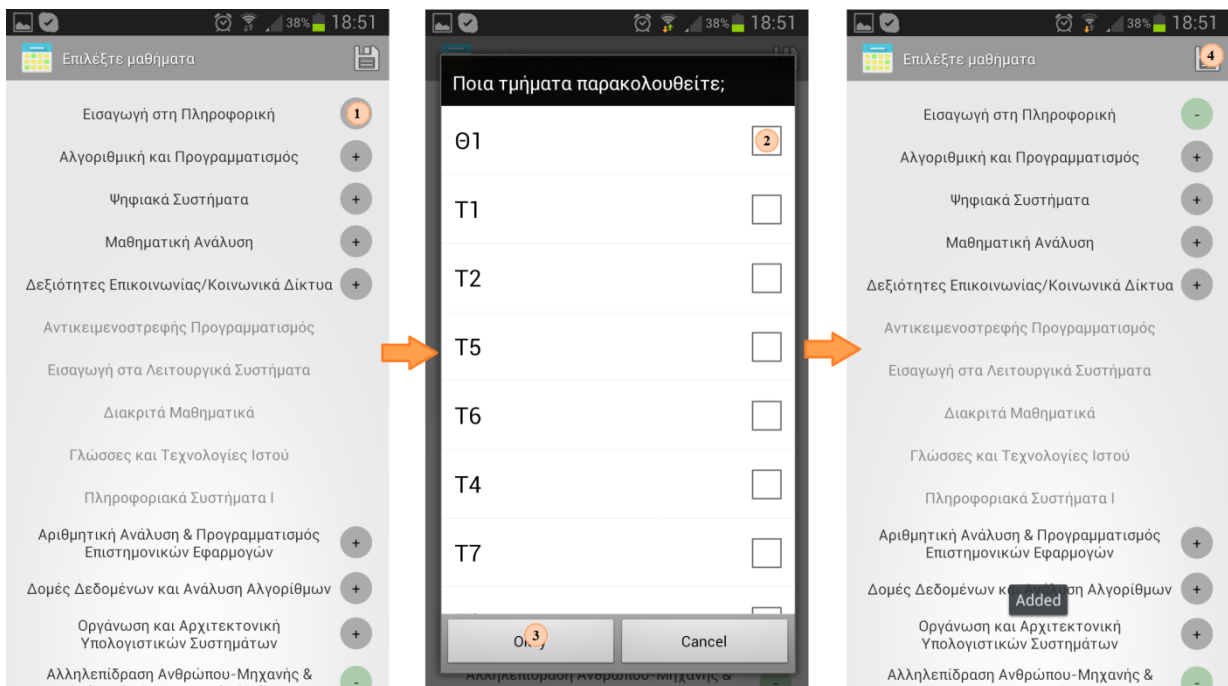
Αυτό το EditModules activity έχει το interface που αναλύθηκε κατά την σχεδίαση και περιέχει όλα τα μαθήματα τα οποία βρίσκει από τον πίνακα μαθηματολόγιο. Αφού είναι η πρώτη εκτέλεση, πρέπει αρχικά να διαβάσει το JSON και να μετατρέψει τα δεδομένα του σε αντικείμενα της γλώσσας Java. Όσην ώρα διαρκεί αυτή η διαδικασία,

η λίστα είναι κενή και εμφανίζεται ένα σύμβολο στην γραμμή ενεργειών που φανερώνει ότι εκτελείται κάποια εργασία όπως φαίνεται στην Εικόνα 25.



Εικόνα 25: Σύμβολο loading κατά το parsing του JSON

Μόλις ολοκληρωθεί η διαδικασία της φόρτωσης των δεδομένων το activity κάνει τις εξής λειτουργίες: δίνει την τιμή true στο disabled για τα μαθήματα που δεν έχουν ούτε μία διάλεξη για το τρέχον εξάμηνο, τους αλλάζει χρώμα, και αποκρύπτει το κουμπί της παρακολούθησής τους. Όταν ο χρήστης πατήσει το κουμπί συν "+" για κάποιο μάθημα, εμφανίζεται ένα αναδυόμενο παράθυρο (popup) το οποίο περιέχει όλα τα τμήματα για το μάθημα αυτό, που υπάρχουν στη λίστα all για το αντίστοιχο μάθημα. Αφού διαλέξει τα τμήματα που επιθυμεί να παρακολουθεί για το συγκεκριμένο μάθημα, το κουμπί αλλάζει σε μείον και γίνεται πράσινο, όπως φαίνεται στην Εικόνα 26.



Εικόνα 26: Βήματα εισαγωγής μαθήματος

Όταν ο χρήστης πατάει το κουμπί Okay στην επιλογή τμημάτων, προστίθενται στην λίστα show τα τμήματα που έχει επιλέξει. Αν η λίστα είναι κενή, το κουμπί ξαναγίνεται γκρι με το σύμβολο συν "+". Τέλος, ολοκληρώνεται η διαδικασία και σώζονται οι προτιμήσεις του χρήστη, μόλις πατήσει το εικονίδιο με την δισκέτα που αντιπροσωπεύει το save. Τότε εμφανίζεται το activity ProboliProgrammatos, το οποίο περιέχει την οθόνη με το ατομικό πρόγραμμα μαθημάτων, όπως σχολιάστηκε κατά τη σχεδίαση. Κάθε γραμμή του πίνακα αυτού δημιουργείται δυναμικά μέσα από το XML με τον ακόλουθο κώδικα.

```
<include layout="@layout/programma_row" android:id="@+id/ora_08me09" />
```

Δυναμικά, επίσης, με την χρήση της μεθόδου inflate δημιουργείται και κάθε συσχέτιση ώρας με μάθημα σύμφωνα με τον παρακάτω κώδικα:

```
View sisxetisiView =
getLayoutInflater().inflate(R.layout.sisxetisi_ora_mathima, cellView,
false);
```

```
[...]
```

```
// add the view to the main layout
```

```
cellView.addView(sisxetisiView);
```

Στη συνέχεια ελέγχει όλο το πρόγραμμα για κάθε μέρα, ώρα και συσχέτιση, και αν το τμήμα του μαθήματος βρίσκεται στην λίστα show, δημιουργεί την αντίστοιχη εγγραφή στο interface, σύμφωνα με την γραμμή.

```
if(sisxetisiOraMathima.getMathima().toShow(sisxetisiOraMathima.getMima()))
```

Στο ίδιο κομμάτι κώδικα γίνεται αρχικά ενημέρωση για την μέρα και ώρα του συστήματος και λαμβάνονται κατάλληλα μέτρα ώστε να εμφανίσει ένα μαύρο περίγραμμα στα μαθήματα που πραγματοποιούνται μέσα στην τρέχουσα ώρα με τον εξής τρόπο:

```
// get Calendar day and match the integer to Meres index
```

```
int currentDay = Calendar.getInstance().get(Calendar.DAY_OF_WEEK);
```

```
if(currentDay<= 1 || currentDay>= 7){
```

```
    currentDay = -1;
```

```
}
```



```

else {
    currentDay = currentDay-2;
}
// get Calendar time and match it to Ores index
int currentTime = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);
if (currentTime>=8 && currentTime<=17){
    currentTime = currentTime-8;
}
else {
    currentTime = -1;
}
[...]
if(currentDay == meresIndex && currentTime == oresIndex){
    cellView.setBackgroundResource(R.color.CURRENT_DAY);
}

```

Το αποτέλεσμα φαίνεται στην Εικόνα 27.

Ωρα	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασ...
	Ψαρράς Νικόλα...				
11-12	Ανάπτ... και Διαχειί... Ολοκλ... Πλ. Συστη... & Εφαρμ... Ζ' 4701 Τ1 301 Σαλαμ... Μιχαήλ Ειδ. Θέματα Δικτύ... Ι Ζ' 4804 Τ1 210 Ψαρράς Νικόλα...	Ψηφιακά Συστή... Α" 4103 Θ1 109 Κλεφτ... Δημήτ... Ανάπτ... Διαδικ... Συστη... & Εφαρμ... Ε' 4504 Τ1 201 Διαμαν... Κωνστ...		Γραφικά Υπολο... Ζ' 4808 Θ1 208 Ράπτης Πασχά...	
		Συστη... & Εφαρμ... Ζ' 4701 Τ2 301 Σαλαμ...			

Εικόνα 27: Περίγραμμα στα μαθήματα που πραγματοποιούνται την τρέχουσα ώρα

Για να μην κόβονται απερίσκεπτα οι λέξεις του τίτλου και του διδάσκοντα του μαθήματος, δημιουργείται δυναμικά μία σειρά για κάθε λέξη μέσα στο πλαίσιο και αν η λέξη είναι μεγαλύτερη του πλαισίου, δεν συνεχίζει στην επόμενη σειρά, αλλά απλά προσθέτονται αποσιωπητικά "..." με τον εξής κώδικα.

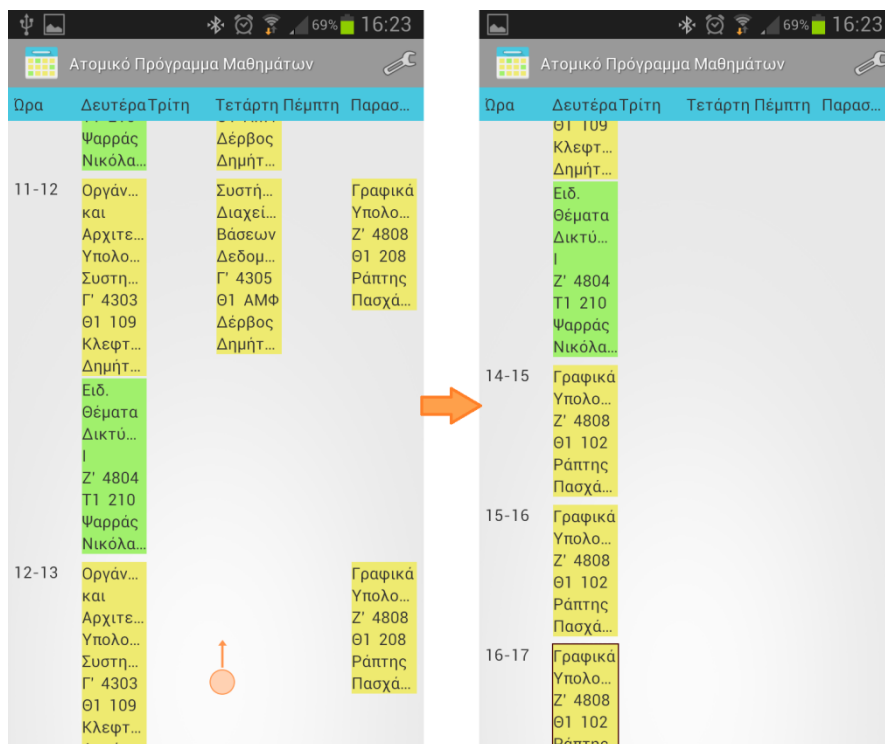
```
String[] titleRows = sisxetisiOraMathima.getMathima().getTitle().split(" ");
    for(int i=0; i<titleRows.length; i++){
        TextView titleRow = new TextView(this);
        titleRow.setEllipsize(TextUtils.TruncateAt.END);
        titleRow.setSingleLine();
        titleRow.setText(titleRows[i]);
        title.addView(titleRow);
    }
```

Επίσης ανάλογα τον τύπο του μαθήματος (εργαστήριο ή θεωρία) αλλάζει το χρώμα του πλαισίου σε πράσινο ή κίτρινο.

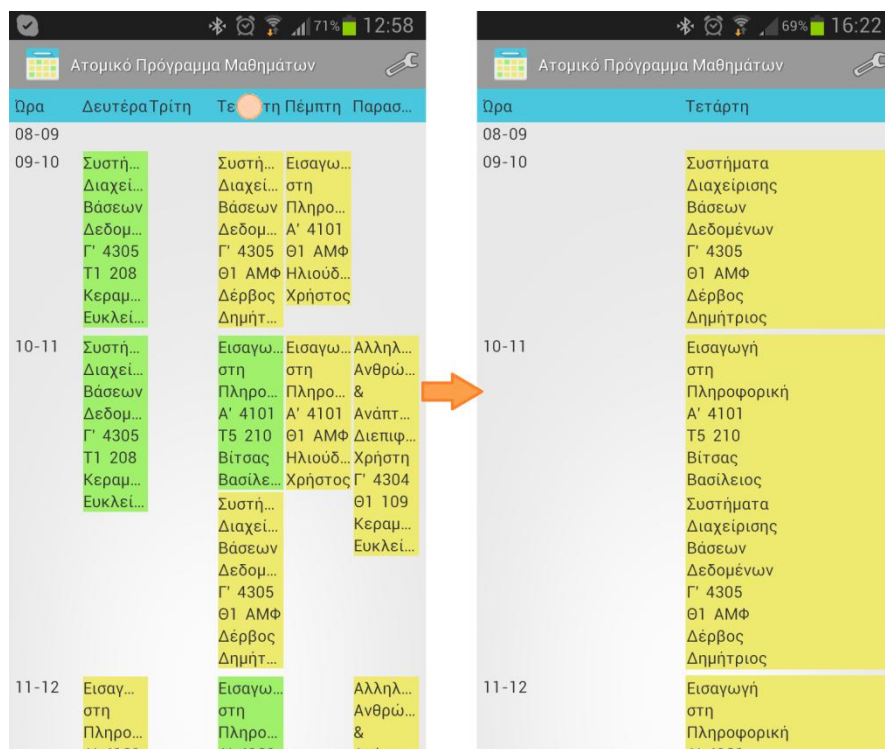
```
if(((int) sisxetisiOraMathima.getTipos()) == 69) {
    sisxetisiView.findViewById(R.id.placeholder).
    setBackgroundResource(R.color.ERGASTIRIO);
}
else if(((int) sisxetisiOraMathima.getTipos()) == 920){
    sisxetisiView.findViewById(R.id.placeholder).
    setBackgroundResource(R.color.THEORIA);
}
```

Παρατηρούμε ότι τα χρώματα βρίσκονται δηλωμένα στα resources στο αρχείο με όνομα color.xml και καλούνται με το όνομα που έχουμε δώσει αντί να χρειάζεται να ορίζονται μέσα στον κώδικα.

Στο activity ProboliProgrammatos ο χρήστης μπορεί να δει ποιες μέρες και ώρες πραγματοποιούνται τα μαθήματα που έχει επιλέξει κάνοντας scroll στην οθόνη ή να απομονώσει τα μαθήματα που πραγματοποιούνται μια συγκεκριμένη μέρα πατώντας πάνω στην ημέρα αυτή όπως φαίνεται στις εικόνες Εικόνα 28 και Εικόνα 299.



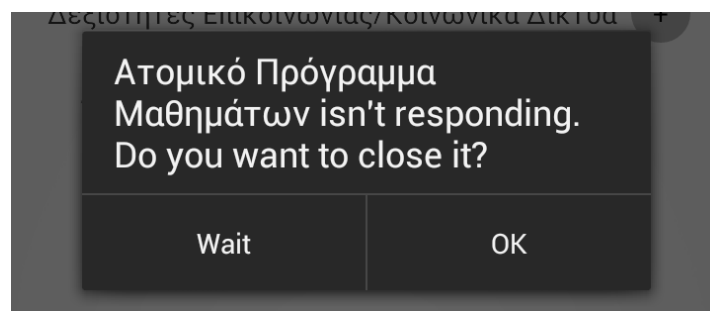
Εικόνα 28: Scrolling



Εικόνα 29: Προβολή συγκεκριμένης ημέρας

3.6 Threads - AsyncTask

Στο Android, μπορούν να αποφευχθούν οι κλάσεις των threads με τη χρήση του αντικειμένου AsyncTask που επιτρέπει την εκτέλεση των πιο σύντομων (διάρκειας μερικών δευτερολέπτων) από τις χρονοβόρες διαδικασίες χωρίς την επιβάρυνση του UI (User Interface). Το αντικείμενο AsyncTask τρέχει τους υπολογισμούς και τις πράξεις σε διαφορετικό thread στο background, ενώ επιστρέφει μόνο το αποτέλεσμα στο thread του UI. Όταν το βασικό thread του UI εκτελεί χρονοβόρες διαδικασίες, το σύστημα τις διακόπτει και εμφανίζει το παρακάτω γνωστό μήνυμα (Εικόνα 30) στον χρήστη, που ενημερώνει για την καθυστέρηση αυτή.



Εικόνα 30: Μήνυμα τερματισμού εφαρμογής λόγω καθυστέρησης στο Main Thread

Για το λόγο αυτό συνιστάται η χρήση του AsyncTask που έχει ως αποτέλεσμα να μην διακόπτεται η εκτέλεση της εφαρμογής όταν διαρκεί περισσότερο από όσο αφήνει το σύστημα. Το AsyncTask χρησιμοποιείται μέσα στην εφαρμογή ορίζοντας το και υπερφορτώνοντας τις εξής μεθόδους του.

Η μέθοδος `onPreExecute()` τρέχει άμεσα στο thread του UI μόλις κληθεί η AsyncTask. Συνήθως κάνει τις απαραίτητες αλλαγές που δηλώνουν την εκτέλεση του AsyncTask, όπως για παράδειγμα να εμφανίσει ένα progress bar (Google Inc).

Η μέθοδος `doInBackground(Params...)` εκτελείται αμέσως μετά την `onPreExecute()` σε thread στο background, και είναι αυτή που κάνει όλη την χρονοβόρα διαδικασία. Η παράμετρος `Params` περιέχει όλα τα στοιχεία που δίδονται στο AsyncTask πριν την εκτέλεση. Αν χρειάζεται, η `doInBackground(Params...)` μπορεί να καλεί την `publishProgress(Progress...)` (Google Inc).

Η `publishProgress(Progress...)` εκτελείται στο `thread` του `UI`, όταν κληθεί από την `doInBackground(Params...)` και χρησιμοποιείται για να εμφανίσει συνήθως την πρόοδο του `AsyncTask` (Google Inc).

Τέλος, μόλις ολοκληρωθεί η `doInBackground(Params...)` εκτελείται στο `main thread` του `UI` η `onPostExecute(Result)` που περιέχει τα αποτελέσματα ως παράμετρο, και τα εμφανίζει όπως ορίζεται στην υπερφόρτωσή της (Google Inc).

Στην προκειμένη εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων" η μόνη χρονοβόρα διαδικασία που χρειάστηκε να αποκοπεί από το `main thread` ήταν αυτή που τελείται μόνο κατά την πρώτη εκτέλεση, και αρχικοποιεί τα αντικείμενα. Χρησιμοποιήθηκαν μόνο οι υπερφορτώσεις της `doInBackground(Params...)` για την ανάγνωση του `JSON` και δημιουργία των αντικειμένων, και της `onPostExecute(Result)` για την προβολή των μαθημάτων στο `activity EditModules` αμέσως μετά την ολοκλήρωση της αρχικοποίηση των αντικειμένων.

3.7 *Serializable* - Αποθήκευση δεδομένων

Όπως είναι λογικό για κάθε εφαρμογή, πρέπει κάπως να αποθηκεύονται τα δεδομένα ώστε να αποφεύγεται να εκτελούνται κάποιες διαδικασίες κάθε φορά που ξαναεκτελείται ο κώδικας. Επιπλέον, όταν ο χρήστης αλληλεπιδρά με την εφαρμογή, συνήθως επιθυμεί να σώσει κάποιες από τις προτιμήσεις του, αφού δεν είναι καθόλου λειτουργικό να πρέπει να τις ξαναδηλώνει κάθε φορά που την ξανατρέχει.

Επίσης επειδή μία από τις ιδιαιτερότητες του `Android` είναι ότι κάθε `activity` είναι ανεξάρτητο από τα υπόλοιπα, για να μεταφερθούν δεδομένα από το ένα `activity` στο άλλο, πρέπει να περιέχονται μέσα στο `intent` που είναι υπεύθυνο για την μετάβαση στο νέο `activity`. Τα αντικείμενα ως έχουν, δεν είναι εφικτό να μεταφερθούν στο νέο `activity` με την μέθοδο `putExtra(...)` των `intents`, γιατί καμία υλοποίηση της δεν δέχεται ως παράμετρο αντικείμενα.

Για τους λόγους αυτούς, στην προκειμένη εφαρμογή χρησιμοποιείται το `Java Serializable interface`. Οποιοδήποτε αντικείμενο η κλάση υλοποιεί τη διεπαφή `Serializable` έχει την δυνατότητα μεταφέρεται από `activity` σε `activity` μέσω της μεθόδου `putExtra(String name, Serializable value)` ή ακόμα και να αποθηκεύεται στην μόνιμη μνήμη της συσκευής ως αρχείο. Ουσιαστικά με τον τρόπο αυτό το αντικείμενο

αναπαριστάται ως ένα σύνολο από δυαδικές ακολουθίες, οι οποίες μεταφέρονται εύκολα ως συμβολοσειρές και μπορούν να γραφτούν σε αρχείο.

Το μόνο ελάττωμα του Serializable είναι ότι αν η κλάση του αντικειμένου που το υλοποιεί τροποποιηθεί, μετά πρέπει να ληφθούν κατάλληλα μέτρα από τον προγραμματιστή ώστε να αποκωδικοποιηθεί σωστά το αρχείο που περιλαμβάνει την οποιαδήποτε προηγούμενη έκδοση του αντικειμένου. Δηλαδή, ακόμα και μία αλλαγή ενός ονόματος πεδίου της κλάσης του αντικειμένου θα δυσκόλευε την αποκωδικοποίησή του. Από τη στιγμή όμως που όλα τα αντικείμενα της εφαρμογής βρίσκονται στην τελική τους μορφή και δεν πρόκειται να αλλάξουν, η αποθήκευσή τους δεν αποτελεί πρόβλημα.

Για την αποθήκευση των αντικειμένων της εφαρμογής δημιουργήθηκε η κλάση Cacher που διαχειρίζεται την εγγραφή τους σε αρχείο, και την ανάγνωσή τους από αυτό. Προϋπόθεση της ορθής λειτουργίας της κλάσης αυτής είναι φυσικά όλα τα αντικείμενα που χρησιμοποιούν τις μεθόδους της να υλοποιούν το interface Serializable. Η κλάση Cacher περιέχει τις εξής στατικές μεθόδους:

```
public static boolean writeObjectToFile(Context context, Object object, String filename) {...}
public static Object readObjectFromFile(Context context, String filename) {...}
```

Όπως φαίνεται από τις υπογραφές τους, η writeObjectToFile χρησιμοποιείται για να γραφτούν τα αντικείμενα στο αρχείο, και δέχεται ως παραμέτρους το activity που την κάλεσε, το προς αποθήκευση αντικείμενο, και το όνομα που πρόκειται να έχει το αρχείο. Αντίστοιχα, η readObjectFromFile δέχεται ως παραμέτρους το activity που την καλεί και το όνομα του προς ανάγνωση αρχείου, ενώ επιστρέφει το αντικείμενο που διαβάζει σε μορφή Object. Το αντικείμενο αυτό στη συνέχεια μπορεί να υποστεί down casting σε οποιοδήποτε άλλο τύπο υλοποιεί.

Στην προκειμένη εφαρμογή τα αντικείμενα που χρειάζεται να σωθούν είναι τα Programma και Mathimatologio. Επειδή, όπως αναλύθηκε στο διάγραμμα κλάσεων, αυτά τα δύο αντικείμενα περιέχουν αντικείμενα τύπου Mathima είναι απαραίτητο οι αναφορές για κάθε Mathima να μένουν ίδιες και στο Programma και στο Mathimatologio. Παρατηρήθηκε ότι αν τα δύο αυτά αντικείμενα δεν αποθηκευτούν ταυτόχρονα, χάνεται η ακεραιότητα των αναφορών επειδή δημιουργούνται δύο στιγμιότυπα για το Mathima. Δηλαδή όταν ο χρήστης πήγαινε να τροποποιήσει την

λίστα των μαθημάτων που επιθυμεί να παρακολουθεί, οι αλλαγές του δεν φαινόταν στο πρόγραμμα, διότι άλλαζε το πεδίο show του μαθήματος, μόνο για το αντικείμενο Mathimatologio, ενώ το Programma έβλεπε πλέον σε νέο στιγμιότυπο Mathima.

Η λύση σε αυτό το πρόβλημα είναι η δημιουργία ενός αντικειμένου δέματος, το ProgrammaBundle, που θα περιέχει το Programma και το Mathimatologio ώστε να μην διαχωριστούν οι αναφορές τους στα μαθήματα. Έτσι η εφαρμογή συνεχίζει να έχει το επιθυμητό αποτέλεσμα χωρίς ο χρήστης να αντιλαμβάνεται ότι με την εναλλαγή των activities τα δεδομένα μεταφέρονται σε νέα αντικείμενα.

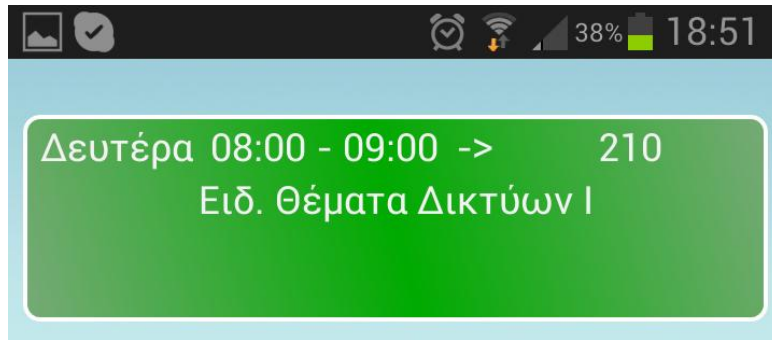
Το αντικείμενο ProgrammaBundle αποθηκεύεται μόνιμα στην μνήμη της συσκευής με την μέθοδο writeObjectToFile, που αναλύθηκε προηγουμένως, έχοντας ως παράμετρο filename το String "ProgrammaBundle". Το αρχείο αυτό αποθηκεύεται μέσα στον φάκελο της εφαρμογής, με το όνομα που ορίστηκε, στο path "/data/data/gr.aetos.programmamathimaton/files/programmaBundle". Σε περίπτωση που η συσκευή στην οποία εκτελείται η μέθοδος αυτή είναι rooted, η διαδρομή αυτή είναι προσπελάσιμη. Το αρχείο ωστόσο, αποτελείται από ακολουθίες byte, που δεν είναι αναγνώσιμες από άνθρωπο. Τα δεδομένα όμως που αποθηκεύονται με αυτόν τον τρόπο μπορούν να σβηστούν από την εφαρμογή "Διαχείριση εφαρμογών" με την επιλογή Clear Data.

3.8 Widget

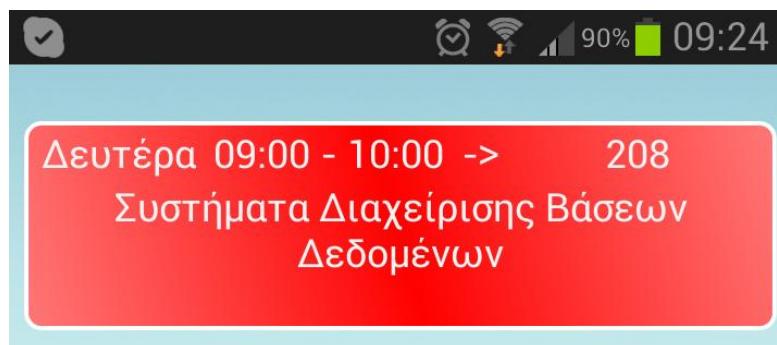
Για την επέκταση της λειτουργικότητας της εφαρμογής θεωρήθηκε χρήσιμο να δημιουργηθεί και ένα widget που θα δείχνει το μάθημα της τρέχουσας ώρας. Σε περίπτωση που δεν πραγματοποιείται κανένα μάθημα εκείνη τη στιγμή, θα εμφανίζει το αμέσως επόμενο μάθημα που θα βρίσκει στο πρόγραμμα.

Το widget (WIndows gaDGET) είναι μια ανεξάρτητη μικρογραφία της εφαρμογής που μπορεί να τοποθετήσει ο χρήστης στην επιφάνεια εργασίας της συσκευής του. Σε αυτό συνήθως εμφανίζονται οι πιο σημαντικές πληροφορίες της εφαρμογής, ώστε να υπάρχει άμεση ενημέρωση του χρήστη. Για την εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων" ένα widget που θα υπενθυμίζει στον φοιτητή-χρήστη ότι έχει μάθημα σύντομα είναι απαραίτητο.

Το layout του widget πρέπει να περιέχει όλες τις απαραίτητες πληροφορίες που χρειάζεται να ξέρει ο χρήστης για το μάθημα που πραγματοποιείται πιο σύντομα. Αυτές οι πληροφορίες είναι η μέρα, ώρα και αίθουσα που πραγματοποιείται το μάθημα, καθώς και ο τίτλος του. Επίσης, θεωρήθηκε χρήσιμο να προστεθεί μια ένδειξη που θα φανερώνει αν πλησιάζει η ώρα έναρξης του μαθήματος. Το αποτέλεσμα της σχεδίασης φαίνεται στις παρακάτω εικόνες Εικόνα 31 και Εικόνα 32.



Εικόνα 31: Το widget όταν το μάθημα δεν πραγματοποιείται σύντομα



Εικόνα 32: Το widget όταν πλησιάζει ή διανύεται η ώρα παρακολούθησης του μαθήματος

Για την υλοποίηση του widget προσπελαύνονται όλοι οι πίνακες του αντικειμένου Programma με σκοπό να βρεθεί το πρώτο μάθημα που περιέχεται στη λίστα show, ξεκινώντας από την ώρα και μέρα που γίνεται η ανανέωσή του widget. Αν δεν βρεθεί μάθημα στην λίστα show την τρέχουσα ώρα, ελέγχεται η επόμενη ώρα. Αν βρεθεί μάθημα στις τρεις πρώτες ώρες από την τρέχουσα ώρα, το widget παίρνει κόκκινο χρώμα. Αλλιώς, ελέγχονται οι επόμενες ώρες της μέρας. Αν δεν πραγματοποιείται μάθημα την τρέχουσα μέρα, προσπελαύνει την επόμενη μέρα, μέχρις ότου βρεθεί μάθημα στην λίστα show.

Το ελάττωμα των widgets είναι ότι με την απλή δημιουργία τους ορίζεται μία περίοδος ανανέωσης τους, η οποία όμως ξεκινά μόλις αυτά τοποθετηθούν στην επιφάνεια

εργασίας, ή ενεργοποιηθεί η συσκευή. Δηλαδή, αν ο χρήστης-φοιτητής αποφασίσει να εισάγει το widget στην επιφάνεια εργασίας της συσκευής του στις 10:55, ενώ το πρώτο του μάθημα ξεκινάει στις 13:00, το widget θα ενημερωθεί για το μάθημα στις 13:55. Το πρόβλημα αυτό λύνεται με την χρήση του συστατικού service.

3.9 Service

Ενώ το συστατικό broadcast receiver ενημερώνεται κάθε ορισμένη περίοδο μετά την δημιουργία του, μπορεί να προγραμματίσει ένα service να εκτελείται κάθε ώρα ακριβώς, σύμφωνα με τον κώδικα.

```
final AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);

final Calendar time = Calendar.getInstance();
time.set(Calendar.MINUTE, 0);
time.set(Calendar.SECOND, 0);
time.set(Calendar.MILLISECOND, 0);

final Intent intent = new Intent(context, MyService.class);

if(service == null){
    System.out.println("service == null");
    service = PendingIntent.getService(context, 0, intent, PendingIntent.FLAG_CANCEL_CURRENT);
}

alarmManager.setRepeating(AlarmManager.RTC, time.getTime().getTime(), AlarmManager.INTERVAL_HOUR, service);
```

Έτσι καλείται ένα service που ενημερώνει το widget κάθε ώρα ακριβώς, ανεξάρτητα από το πότε το widget τοποθετήθηκε στην επιφάνεια εργασίας. Επίσης το widget ενημερώνεται όταν ο χρήστης αποθηκεύσει τις προτιμήσεις του, καλώντας το ίδιο service μέσω του activity EditModules, ώστε να περιέχει το νέο πρόγραμμα που έχει δηλωθεί. Τέλος, το widget είναι και συντόμευση του προγράμματος, αφού όπου και να πατήσει ο χρήστης πάνω σε αυτό, τον μεταφέρει στο activity ProboliProgrammatos, χάρη στον κατάλληλο χειρισμό του onClick event.

3.10 Επίλογος

Σε αυτό το κεφάλαιο, αναλύθηκε η εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων", ξεκινώντας από τους στόχους και τις απαιτήσεις της, μέχρι το interface και τον κώδικα. Αναφέρθηκαν επίσης τα χαρακτηριστικά που την βελτιώνουν, όπως το AsyncTask, το Serializable Interface, το widget και το service. Επομένως, μπορεί πλέον να περιγραφεί η τεχνολογία NFC που θα μπορούσε να χρησιμοποιηθεί για την επέκτασή της.

ΚΕΦΑΛΑΙΟ 4: Τεχνολογία Near Field Communication (NFC)

Όπως αναφέρθηκε στο υποκεφάλαιο 1.3 παραπάνω της ιστορικής αναδρομής του λειτουργικού Android, από την έκδοση Gingerbread άρχισε να υποστηρίζεται η τεχνολογία Near Field Communication (NFC) (Λογότυπο στην Εικόνα 33). Πρόκειται για ένα σύνολο προτύπων για μετάδοση και λήψη πληροφοριών σε πολύ κοντινές αποστάσεις, σχεδόν επαφής. Σε αυτό το κεφάλαιο θα περιγραφεί η τεχνολογία αυτή και θα αναφερθούν τα προβλήματα ασφαλείας που δημιουργούνται. Τέλος, θα αναφερθούν εφαρμογές που την υλοποιούν, καθώς και η σημαντική επέκταση που θα μπορούσε να έχει στην εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων".



Εικόνα 33: Λογότυπο NFC

4.1 Χαρακτηριστικά

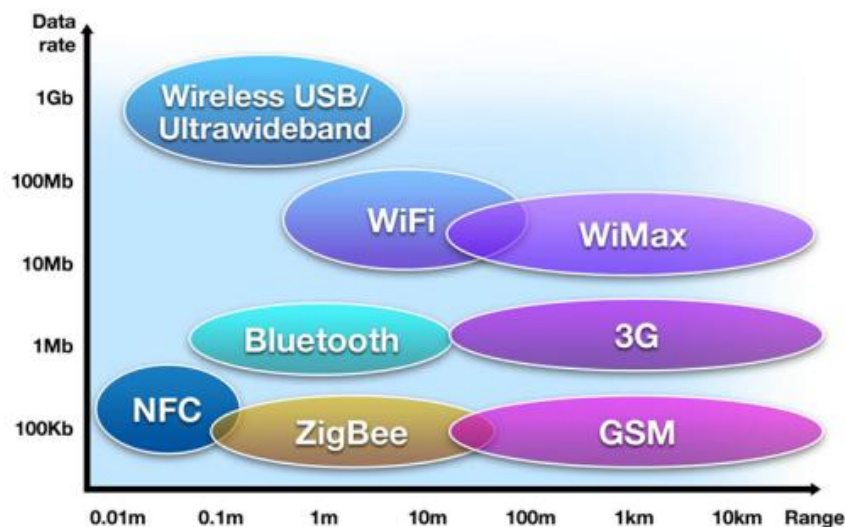
Το NFC βασίστηκε στο Radio-Frequency Identification (RFID) το οποίο χρησιμοποιείται για αυθεντικοποίηση χρηστών ή εντοπισμό αντικειμένων. Η κύρια διαφορά τους είναι ότι η τεχνολογία RFID υποστηρίζει μονόδρομη επικοινωνία, ενώ το NFC υποστηρίζει αμφίδρομη επικοινωνία (Wikipedia).

Η συχνότητα στην οποία μεταδίδει το NFC είναι 13.56 MHz (μήκος κύματος 22 μέτρα). Λόγω της δυσαναλογίας διαστάσεων με τις κεραίες των συσκευών που το υποστηρίζουν έχει ως αποτέλεσμα την μικρή εμβέλεια των 0 ως 5 εκατοστών. Συνήθως αυτή η κεραία βρίσκεται στην εξωτερική πλευρά της μπαταρίας, ώστε να "βλέπει" την απέναντι συσκευή (Antenna Theory .com). Η μπαταρία περιβάλλεται από κάλυμμα που δεν εμποδίζει την ακτινοβολία και αν αφαιρεθεί φαίνεται όπως στην Εικόνα 34.



Εικόνα 34: Κεραία NFC στη μπαταρία συσκευής

Η ταχύτητα μετάδοσης μέσω NFC είναι πολύ μικρότερη από αυτή των υπολοίπων ασυρμάτων επικοινωνιών φτάνοντας μόνο μέχρι τα 424Kbps, όπως φαίνεται από την Εικόνα 35 (NFC Forum).



Εικόνα 35: Ταχύτητες και εμβέλειες ασυρμάτων επικοινωνιών

Το NFC αποκαθιστά επικοινωνία πολύ γρήγορα (λιγότερο από 100ms), σε αντίθεση με άλλες ασύρματες τεχνολογίες. Για τον λόγο αυτό χρησιμοποιείται και ως αρχικό στάδιο ζεύξης για άλλες ασύρματες επικοινωνίες μεταξύ συσκευών (Wikipedia).

Λόγω της διάταξης του ηλεκτρονικού κυκλώματος πομπού - δέκτη, ο δέκτης δεν χρειάζεται απαραίτητα να είναι συνδεδεμένος με ηλεκτρική πηγή, διότι μπορεί να

τροφοδοτείται επαγωγικά μέσω του πομπού. Έτσι διακρίνουμε δύο βασικές κατηγορίες: την ενεργητική κατηγορία και στην παθητική κατηγορία (Wikipedia).

Στην ενεργητική κατηγορία η συσκευή Android συνδέεται με άλλη τροφοδοτούμενη συσκευή, όπως μηχανήματα πληρωμών, ή άλλες συσκευές Android. Σε αυτήν την περίπτωση, η κατανάλωση του πομπού είναι μικρότερη από την κατανάλωση που θα είχε χρησιμοποιώντας Bluetooth. Στην παθητική κατηγορία η συσκευή Android επικοινωνεί με tags, όπως αυτοκόλλητα NFC, έξυπνες αφίσες NFC, έξυπνες κάρτες (smartcards) εφόσον περιέχουν κεραία NFC, μπρελόκ NFC, που δεν διαθέτουν δική τους τροφοδοσία. Ως αποτέλεσμα, η κατανάλωση στην πλευρά του πομπού αυξάνεται ελαφρώς (Wikipedia).

4.2 Ασφάλεια

Παρότι το NFC λειτουργεί σε πολύ μικρές αποστάσεις, από μόνο του δεν μπορεί να εγγυηθεί μια ασφαλή επικοινωνία. Γίνεται όμως να αποφευχθούν κάποιες μορφές επιθέσεων εάν οι εφαρμογές χρησιμοποιούν πρωτόκολλα κρυπτογράφησης (όπως το SSL) για να συνδεθούν μέσω ασφαλών καναλιών (Wikipedia).

Η εμπιστευτικότητα της επικοινωνίας NFC καταπατάται με επιθέσεις eavesdropping. Δηλαδή, το σήμα της ασύρματης επικοινωνίας μπορεί να υποκλαπεί μέσω κεραιών. Ανάλογα με την κατηγορία, παθητική ή ενεργητική, η μη εξουσιοδοτημένη κεραία μπορεί να απέχει 1 μέτρο ή 10 μέτρα αντίστοιχα, ενώ δεν είναι απαραίτητη η φυσική παρουσία αυτού που επιτίθεται (Wikipedia).

Η χρήση μηχανημάτων που αλλοιώνουν την επικοινωνία του εξουσιοδοτημένου πομπού και δέκτη προσβάλουν την ακεραιότητα, εμπιστευτικότητα και διαθεσιμότητα της επικοινωνίας. Αυτή η επίθεση δεν υπάρχει τρόπος να αποτραπεί, αλλά μπορεί να γίνει αντιληπτή σε περιπτώσεις που γίνεται έλεγχος των μηνυμάτων που λαμβάνονται.

Άλλες μορφές επιθέσεων συμβαίνουν όταν ο επιτιθέμενος παρεμβάλλεται μεταξύ του νόμιμου πομπού και δέκτη δίνοντας τους την εντύπωση πως αυτός είναι ο νόμιμος παραλήπτης. Επίσης, συμβαίνουν όταν βρεθεί στην κατοχή του επιτιθέμενου η συσκευή του νόμιμου πομπού, ή όταν ο επιτιθέμενος διατηρήσει το κανάλι επικοινωνίας πομπού - δέκτη ανοιχτό μετά την απομάκρυνση ενός από τους δύο.

4.3 Εφαρμογές

Η τεχνολογία NFC άρχισε να γίνεται γνωστή στις Android συσκευές με την χρήση του Android Beam και του S Beam που χρησιμοποιούνται για την ανταλλαγή αρχείων. Το Android Beam ουσιαστικά διεκπεραιώνει την επικοινωνία μέσω Bluetooth μεταξύ δύο Android συσκευών κάνοντας την διαδικασία "χειραφίας" μέσω NFC. Αντίστοιχα και το S Beam (που υποστηρίζεται αντί του Android Beam από συσκευές Samsung) αποκαθιστά την Wi-Fi επικοινωνία των δύο συσκευών ανταλλάσσοντας το IP address και το MAC address μέσω του NFC.

Επίσης, το NFC χρησιμοποιείται και στις συναλλαγές με τη χρήση ειδικών NFC μηχανημάτων στα ταμεία όπως αυτού της Εικόνα 36. Ο χρήστης συνδέει την Android συσκευή του με το Google Wallet, ή άλλο ηλεκτρονικό πορτοφόλι, και πληρώνει απλά πλησιάζοντάς την στο κατάλληλο μηχάνημα, με το ειδικό NFC σύμβολο.



Εικόνα 36: Ολοκλήρωση συναλλαγής μέσω NFC

Άλλη εφαρμογή της τεχνολογίας αυτής συναντάμε με την χρήση των tags. Τα tags χρησιμεύουν για την προσπέλαση ενός απλού URL ή μίας συμβολοσειράς, ως την εκτέλεση μιας σειράς εντολών, μόλις η Android συσκευή τα πλησιάσει. Χωρίζονται σε τέσσερις κατηγορίες ανάλογα με την ταχύτητα επικοινωνίας, τη δυνατότητα

παραμετροποίησης, την ασφάλεια των δεδομένων και τη χωρητικότητα μνήμης που κυμαίνεται από 96 ως 4,096 bytes (Wikipedia). Ανάλογα με τις απαιτήσεις της εφαρμογής, χρησιμοποιείται και το tag της κατάλληλης κατηγορίας. Ένα παράδειγμα αυτοκόλλητου tag φαίνεται στην Εικόνα 37.



Εικόνα 37: Αυτοκόλλητο NFC tag

4.4 Επέκταση της εφαρμογής με τη χρήση του NFC

Στα πλαίσια αυτής της εργασίας, η τεχνολογία NFC θα μπορούσε να αποτελέσει μία σημαντική επέκταση της εφαρμογής "Ατομικό Πρόγραμμα Μαθημάτων". Αυτό θα συνέβαινε εάν τοποθετούνταν NFC tags έξω από κάθε αίθουσα διδασκαλίας του τμήματος. Έτσι, ύστερα από κατάλληλη ρύθμιση της εφαρμογής και των tags, κάθε φοιτητής - χρήστης της εφαρμογής, θα μπορούσε να πλησιάζει τη συσκευή του σε ένα από αυτά. Η εφαρμογή θα ενημερωνόταν άμεσα για την αίθουσα στην οποία βρίσκεται ο φοιτητής, και θα εμφάνιζε το πρόγραμμα της. Αν ο φοιτητής είχε μάθημα εκείνη τη στιγμή στην συγκεκριμένη αίθουσα, θα τον ενημέρωνε η εφαρμογή, και θα έβαζε αυτόματα τη συσκευή του σε "λειτουργία μαθήματος", απενεργοποιώντας τους ήχους της.

4.5 Επίλογος

Όπως αναλύθηκε στο κεφάλαιο αυτό, η τεχνολογία NFC βρίσκει διάφορες εφαρμογές σε περιπτώσεις που χρειάζεται άμεση επικοινωνία συσκευών σε κοντινές αποστάσεις. Για τον λόγο αυτό θα μπορούσε να επεκτείνει και την εφαρμογή που υλοποιήθηκε για την εργασία αυτή, με προϋπόθεση να ληφθούν όμως και τα κατάλληλα μέτρα ασφαλείας που χρειάζονται. Στο επόμενο κεφάλαιο θα σχολιαστούν τα συμπεράσματα που προκύπτουν από την εργασία αυτή.

Ανασκόπηση - Συμπεράσματα

Στην παρούσα εργασία αναπτύχθηκαν τα θέματα που χρειάζονται για την υλοποίηση της εφαρμογής, όπως το λειτουργικό σύστημα Android και η συμβολική γλώσσα JSON, ενώ μετά περιγράφηκε η εφαρμογή "Ατομικό Πρόγραμμα Μαθημάτων". Τέλος έγινε αναφορά στην τεχνολογία NFC που θα μπορούσε να χρησιμοποιηθεί για την επέκταση της εφαρμογής.

Αρχικά περιγράφηκε το λειτουργικό σύστημα Android. Έγινε δηλαδή μία ιστορική αναδρομή και αναφορά στην αρχιτεκτονική του. Επίσης αναλύθηκαν τα συστατικά τα οποία απαρτίζουν κάθε εφαρμογή, η σημασία της χρήσης XML, καθώς και κάποια θέματα ασφαλείας.

Στο δεύτερο κεφάλαιο παρουσιάστηκε η συμβολική γλώσσα JSON μαζί με την σύνταξή της, και τις χρήσεις της. Επιπλέον εξηγήθηκαν οι λόγοι για τους οποίους προτιμάται.

Το τρίτο κεφάλαιο αφορά την εφαρμογή που υλοποιήθηκε στα πλαίσια της πτυχιακής εργασίας αυτής. Πρόκειται για μία εφαρμογή που προβάλλει το ατομικό πρόγραμμα μαθημάτων των φοιτητών του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Κατά τη σχεδίαση της εφαρμογής, αποφασίστηκε να "διαβάζεται" το πρόγραμμα μαθημάτων του τμήματος από ένα JSON αρχείο και να δημιουργεί τα αντίστοιχα αντικείμενα. Στη συνέχεια ο φοιτητής διαλέγει τα μαθήματα που επιθυμεί να παρακολουθεί, με αποτέλεσμα να ενημερώνονται τα αντικείμενα και να αποθηκεύονται στη μνήμη του κινητού με τη χρήση του Serializable interface. Μόλις ολοκληρωθεί η διαδικασία αυτή, τα μαθήματα που παρακολουθεί ο φοιτητής απεικονίζονται στην οθόνη της συσκευής, μαζί με άλλα σχετικά στοιχεία, σε εβδομαδιαία ή ημερήσια μορφή. Τέλος δίνεται η δυνατότητα ενημέρωσης του φοιτητή - χρήστη μέσω του widget που η εφαρμογή διαθέτει.

Στο τελευταίο κεφάλαιο παρουσιάζεται η τεχνολογία Near Field Communication (NFC) που θα μπορούσε να αποτελέσει σημείο εκκίνησης για περαιτέρω επέκταση της εφαρμογής που υλοποιήθηκε.

Σύμφωνα με όσα περιγράφηκαν στην εργασία αυτή, συμπεραίνεται ότι το λειτουργικό σύστημα Android είναι αρκετά αξιόπιστο, εύχρηστο και ευέλικτο για έξυπνες συσκευές.

Επιπλέον λόγω των αδειών που διαθέτει και επειδή είναι λογισμικό ανοιχτού κώδικα, δίνει στους προγραμματιστές την ευκαιρία να υλοποιήσουν εφαρμογές για αυτό, οι οποίες μπορούν να αποφέρουν κέρδος χωρίς ιδιαίτερο κόπο εισόδου στην αγορά εργασίας.

Επισημαίνεται επίσης ότι είναι αρκετά βολικό για καθηγητές και φοιτητές να υπάρχουν εφαρμογές που να τους ενημερώνουν για τα προγράμματα μαθημάτων τους. Θα μπορούσε λοιπόν, η παρούσα εφαρμογή, να αποτελέσει πηγή έμπνευσης για ένα μεγαλύτερο και συνολικότερο project που θα περιλάμβανε πλήρως τα Ιδρύματα στον ευρύτερο χώρο της Ανώτερης και Ανώτατης Εκπαίδευσης.

Τέλος, λόγω της επαναχρησιμοποίησης κώδικα, διαπιστώνεται ότι η εφαρμογή αυτή διαθέτει όλα τα βασικά στοιχεία που χρειάζεται μια εφαρμογή "Εορτολόγιο". Αυτό συμβαίνει διότι περιέχει μία προβολή προγράμματος που θα μπορούσε να αντικατασταθεί από προβολή ονομαστικών εορτών ανά ημέρα, καθώς και ένα widget που θα μπορούσε να υπενθυμίζει για τους εορτάζοντες την τρέχουσα ημέρα.

Αναφορές και Βιβλιογραφία

Γκούτζος, Ά. (2013). *Ανάπτυξη πλατφόρμας και εφαρμογής κινητού τηλεφώνου για παιχνίδι τύπου «κρυμμένος θησαυρός» ως μέσο ξενάγησης και διαφήμισης*. ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ.

Al Salool, A. *Android Architecture For System Application Software Stack*. Ανακτήθηκε 22 Νοεμβρίου, 2013, από <http://android-app-tutorial.blogspot.gr/2012/08/architecture-system-application-stack.html#.Uo5m3OKJvcs>.

Antenna Theory .com. *NFC Antennas*. Ανακτήθηκε 11 Δεκεμβρίου, 2013, από <http://www.antenna-theory.com/definitions/nfc-antenna.php>.

Crockford, D. *Introducing JSON*. Ανακτήθηκε 5 Νοεμβρίου, 2013, από <http://json.org/>.

Crockford, D. *JSON: The Fat-Free Alternative to XML*. Ανακτήθηκε 5 Νοεμβρίου, 2013, από <http://www.json.org/fatfree.html>.

Darcey, L. & Conder, S. (2010). *Sams Teach Yourself Android Application Development in 24 hours.*: Sams.

Felker, D. *Android Applications Development FOR DUMMIES.*: Wiley Publishing.

Google Inc. *Android Open Source Project*. Ανακτήθηκε 10 Νοεμβρίου, 2013, από <http://source.android.com/>.

Google Inc. *Activities*. Ανακτήθηκε 24 Νοεμβρίου, 2013, από <http://developer.android.com/guide/components/activities.html>.

Google Inc. *App Widgets*. Ανακτήθηκε 4 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/appwidgets/index.html#AppWidgetProvider>.

Google Inc. *Application Fundamentals*. Ανακτήθηκε 19 Νοεμβρίου, 2013, από <http://developer.android.com/guide/components/fundamentals.html#broadlife>.

Google Inc. *AsyncTask*. Ανακτήθηκε 3 Δεκεμβρίου, 2013, από <http://developer.android.com/reference/android/os/AsyncTask.html>.

Google Inc. *BroadcastReceiver*. Ανακτήθηκε 4 Δεκεμβρίου, 2013, από <http://developer.android.com/reference/android/content/BroadcastReceiver.html>.

Google Inc. *Content Providers*. Ανακτήθηκε 1 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/providers/content-providers.html>.

Google Inc. *Content Provider Basics*. Ανακτήθηκε 1 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/providers/content-provider-basics.html>.

Google Inc. *Get the Android SDK*. Ανακτήθηκε 7 Δεκεμβρίου, 2013, από <http://developer.android.com/sdk/index.html>.

Google Inc. *Intent*. Ανακτήθηκε 3 Δεκεμβρίου, 2013, από <http://developer.android.com/reference/android/content/Intent.html>.

Google Inc. *Layouts*. Ανακτήθηκε 6 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/ui/declaring-layout.html>.

Google Inc. *Prices and supported currencies*. Ανακτήθηκε 7 Δεκεμβρίου, 2013, από <https://support.google.com/googleplay/android-developer/answer/138412?hl=en&topic=15867&ctx=topic>.

Google Inc. *Processes and Threads*. Ανακτήθηκε 3 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/components/processes-and-threads.html>.

Google Inc. *Serializable*. Ανακτήθηκε 3 Δεκεμβρίου, 2013, από <http://developer.android.com/reference/java/io/Serializable.html>.

Google Inc. *Services*. Ανακτήθηκε 24 Νοεμβρίου, 2013, από <http://developer.android.com/guide/components/services.html>.

Google Inc. *The AndroidManifest.xml File*. Ανακτήθηκε 6 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/manifest/manifest-intro.html>.

Google Inc. *UI Overview*. Ανακτήθηκε 6 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/ui/overview.html#Layout>.

Google Inc. *Widgets*. Ανακτήθηκε 4 Δεκεμβρίου, 2013, από <http://developer.android.com/design/patterns/widgets.html>.

Google Inc. <uses-sdk>. Ανακτήθηκε 6 Δεκεμβρίου, 2013, από <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>.

NFC Forum. *NFC Forum Technical Specifications*. Ανακτήθηκε 11 Δεκεμβρίου, 2013, από http://www.nfc-forum.org/specs/spec_list/.

The Verge. *Android: A visual history*. Ανακτήθηκε 17 Νοεμβρίου, 2013, από <http://www.theverge.com/2011/12/7/2585779/android-history>.

Visual Paradigm International. *Visual Paradigm Know-how*. Ανακτήθηκε 19 Νοεμβρίου, 2013, από <http://knowhow.visual-paradigm.com/uml/10-use-case-diagram-tips/>.

Visual Paradigm International. *Writing Effective Use Case*. Ανακτήθηκε 19 Νοεμβρίου, 2013, από

<http://www.visual-paradigm.com/product/vpuml/tutorials/writingeffectiveusecase.jsp>.

Android From Wikipedia, the free encyclopedia. Ανακτήθηκε 6 Νοεμβρίου, 2013, από http://en.wikipedia.org/wiki/Android_%28operating_system%29.

Dalvik (software) From Wikipedia the free encyclopedia. Ανακτήθηκε 22 Νοεμβρίου, 2013, από http://en.wikipedia.org/wiki/Dalvik_%28software%29.

Google Play From Wikipedia, the free encyclopedia. Ανακτήθηκε 7 Δεκεμβρίου, 2013, από http://en.wikipedia.org/wiki/Google_Play.

Near field communication From Wikipedia, the free encyclopedia. Ανακτήθηκε 11 Δεκεμβρίου, 2013, από http://en.wikipedia.org/wiki/Near_field_communication.

Smartphone From Wikipedia, the free encyclopedia. Ανακτήθηκε 5 Νοεμβρίου, 2013, από <http://en.wikipedia.org/wiki/Smartphone>.

Strickland, J. *How the Google Phone Works - Google Android Architecture*. Ανακτήθηκε 22 Νοεμβρίου, 2013, από <http://electronics.howstuffworks.com/google-phone2.htm>.

JSON From Wikipedia, the free encyclopedia. Ανακτήθηκε 5 Νοεμβρίου, 2013, από <http://en.wikipedia.org/wiki/JSON>.

Παράρτημα 1: Πηγή δεδομένων JSON

```
[
  {
    "id" = 4101,
    "title" = "Εισαγωγή στη Πληροφορική",
    "eksamino" = "Α",
    sisxetiseisOraMathima = [
      {
        mera: "_DEUTERA",
        ora: "_11me12",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "ΑΜΦ",
        didaskon: "Ηλιούδης Χρήστος"
      },
      {
        mera: "_DEUTERA",
        ora: "_12me13",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "ΑΜΦ",
        didaskon: "Ηλιούδης Χρήστος"
      },
      {
        mera: "_TRITI",
        ora: "_16me17",
        tipos: "Ε",
        tmima: "Τ1",
        aithousa: "210",
        didaskon: "Γιακουσιδης Κωσταντίνος"
      },
      {
        mera: "_TRITI",
        ora: "_17me18",
        tipos: "Ε",
        tmima: "Τ1",
        aithousa: "210",
        didaskon: "Γιακουσιδης Κωσταντίνος"
      },
      {
        mera: "_TETARTI",
        ora: "_08me09",
        tipos: "Ε",
        tmima: "Τ2",
        aithousa: "210",
        didaskon: "Βίτσας Βασίλειος"
      },
      {
        mera: "_TETARTI",
        ora: "_09me10",
        tipos: "Ε",
        tmima: "Τ2",
        aithousa: "210",
        didaskon: "Βίτσας Βασίλειος"
      },
      {
        mera: "_TETARTI",
        ora: "_10me11",
        tipos: "Ε",
        tmima: "Τ5",
        aithousa: "210",
        didaskon: "Βίτσας Βασίλειος"
      },
      {
        mera: "_TETARTI",
        ora: "_10me11",
        tipos: "Ε",
        tmima: "Τ6",
        aithousa: "202",
        didaskon: "Ηλιούδης Χρήστος"
      }
    ]
  }
]
```

```

    mera: "_TETARTI",
    ora: "_11me12",
    tipos: "E",
    tmima: "T5",
    aithousa: "210",
    didaskon: "Βίτσας Βασίλειος"
},
{
    mera: "_TETARTI",
    ora: "_11me12",
    tipos: "E",
    tmima: "T6",
    aithousa: "202",
    didaskon: "Ηλιούδης Χρήστος"
},
{
    mera: "_PEMPTI",
    ora: "_09me10",
    tipos: "0",
    tmima: "01",
    aithousa: "ΑΜΦ",
    didaskon: "Ηλιούδης Χρήστος"
},
{
    mera: "_PEMPTI",
    ora: "_10me11",
    tipos: "0",
    tmima: "01",
    aithousa: "ΑΜΦ",
    didaskon: "Ηλιούδης Χρήστος"
},
{
    mera: "_PEMPTI",
    ora: "_14me15",
    tipos: "E",
    tmima: "T4",
    aithousa: "208",
    didaskon: "Αντωνίου Στάθης"
},
{
    mera: "_PEMPTI",
    ora: "_15me16",
    tipos: "E",
    tmima: "T4",
    aithousa: "208",
    didaskon: "Αντωνίου Στάθης"
},
{
    mera: "_PEMPTI",
    ora: "_16me17",
    tipos: "E",
    tmima: "T7",
    aithousa: "210",
    didaskon: "Γιακουσιδής Κωσταντίνος"
},
{
    mera: "_PEMPTI",
    ora: "_17me18",
    tipos: "E",
    tmima: "T7",
    aithousa: "210",
    didaskon: "Γιακουσιδής Κωσταντίνος"
},
{
    mera: "_PARASKEUI",
    ora: "_11me12",
    tipos: "E",
    tmima: "T3",
    aithousa: "202",
    didaskon: "Γιακουσιδής Κωσταντίνος"
},
{
    mera: "_PARASKEUI",
    ora: "_12me13",

```

```

        tipos: "E",
        tmima: "T3",
        aithousa: "202",
        didaskon: "Γιακουσιδης Κωσταντινος"
    }
]
},{
    "id" = 4102,
    "title" = "Αλγοριθμική και Προγραμματισμός",
    "eksamino" = "A",
    sisxetiseisOraMathima = [
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Σφέτσος Παναγιώτης"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Σφέτσος Παναγιώτης"
        },
        {
            mera: "_TRITI",
            ora: "_14me15",
            tipos: "E",
            tmima: "T1",
            aithousa: "210",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_TRITI",
            ora: "_15me16",
            tipos: "E",
            tmima: "T1",
            aithousa: "210",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_TETARTI",
            ora: "_08me09",
            tipos: "E",
            tmima: "T3",
            aithousa: "208",
            didaskon: "Σφέτσος Παναγιώτης"
        },
        {
            mera: "_TETARTI",
            ora: "_08me09",
            tipos: "E",
            tmima: "T6",
            aithousa: "211",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_TETARTI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T3",
            aithousa: "208",
            didaskon: "Σφέτσος Παναγιώτης"
        },
        {
            mera: "_TETARTI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T6",
            aithousa: "211",
            didaskon: "Γιακουσιδης Κωσταντινος"
        }
    ]
}

```

```

    },
    {
        mera: "_TETARTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T2",
        aithousa: "301",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_TETARTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T2",
        aithousa: "301",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_TETARTI",
        ora: "_16me17",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Γιακουσιδης Κωσταντινος"
    },
    {
        mera: "_TETARTI",
        ora: "_17me18",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Γιακουσιδης Κωσταντινος"
    },
    {
        mera: "_PEMPTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T5",
        aithousa: "202",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_PEMPTI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T5",
        aithousa: "202",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_PARASKEUI",
        ora: "_09me10",
        tipos: "0",
        tmima: "01",
        aithousa: "AM0",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_PARASKEUI",
        ora: "_10me11",
        tipos: "0",
        tmima: "01",
        aithousa: "AM0",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {
        mera: "_PARASKEUI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T7",
        aithousa: "210",
        didaskon: "Σφέτσος Παναγιώτης"
    },
    {

```

```

        mera: "_PARASKEUI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T7",
        aithousa: "210",
        didaskon: "Σφέτσος Παναγιώτης"
    }
]
}, {
    "id" = 4103,
    "title" = "Ψηφιακά Συστήματα",
    "eksamino" = "A",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_DEUTERA",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_12me13",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        }
    ]
}, {
    "id" = 4104,
    "title" = "Μαθηματική Ανάλυση",
    "eksamino" = "A",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_14me15",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Αντωνίου Στάθης"
        },
        {
            mera: "_DEUTERA",
            ora: "_15me16",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Αντωνίου Στάθης"
        },
        {
            mera: "_DEUTERA",
            ora: "_16me17",
            tipos: "0",
            tmima: "01",
            aithousa: "AM0",
            didaskon: "Αντωνίου Στάθης"
        }
    ]
}

```



```

    },
    {
        mera: "_TETARTI",
        ora: "_14me15",
        tipos: "0",
        tmima: "01",
        aithousa: "102",
        didaskon: "Αντωνίου Στάθης"
    },
    {
        mera: "_TETARTI",
        ora: "_15me16",
        tipos: "0",
        tmima: "01",
        aithousa: "102",
        didaskon: "Αντωνίου Στάθης"
    }
]
},{
    "id" = 4105,
    "title" = "Δεξιότητες Επικοινωνίας/Κοινωνικά Δίκτυα",
    "eksamino" = "Α",
    sisxetiseisOraMathima = [
        {
            mera: "_TETARTI",
            ora: "_08me09",
            tipos: "E",
            tmima: "T5",
            aithousa: "202",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_TETARTI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T5",
            aithousa: "202",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_TETARTI",
            ora: "_10me11",
            tipos: "E",
            tmima: "T3",
            aithousa: "211",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_TETARTI",
            ora: "_11me12",
            tipos: "E",
            tmima: "T3",
            aithousa: "211",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_PEMPTI",
            ora: "_11me12",
            tipos: "E",
            tmima: "T6",
            aithousa: "201",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_PEMPTI",
            ora: "_11me12",
            tipos: "E",
            tmima: "T1",
            aithousa: "211",
            didaskon: "Γιακουσιδης Κωσταντινος"
        },
        {
            mera: "_PEMPTI",
            ora: "_12me13",

```

```

        tipos: "E",
        tmima: "T6",
        aithousa: "201",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PEMPTI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T1",
        aithousa: "211",
        didaskon: "Γιακουσιδης Κωσταντινος"
    },
    {
        mera: "_PEMPTI",
        ora: "_14me15",
        tipos: "E",
        tmima: "T2",
        aithousa: "301",
        didaskon: "Γιακουσιδης Κωσταντινος"
    },
    {
        mera: "_PEMPTI",
        ora: "_14me15",
        tipos: "E",
        tmima: "T7",
        aithousa: "211",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PEMPTI",
        ora: "_15me16",
        tipos: "E",
        tmima: "T2",
        aithousa: "301",
        didaskon: "Γιακουσιδης Κωσταντινος"
    },
    {
        mera: "_PEMPTI",
        ora: "_15me16",
        tipos: "E",
        tmima: "T7",
        aithousa: "211",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PEMPTI",
        ora: "_16me17",
        tipos: "E",
        tmima: "T4",
        aithousa: "211",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PEMPTI",
        ora: "_17me18",
        tipos: "E",
        tmima: "T4",
        aithousa: "211",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PARASKEUI",
        ora: "_14me15",
        tipos: "0",
        tmima: "01",
        aithousa: "ΑΜΦ",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PARASKEUI",
        ora: "_15me16",
        tipos: "0",
        tmima: "01",
    }

```

```

        aithousa: "ΑΜΦ",
        didaskon: "Σιάκα Κέρστιν"
    },
    {
        mera: "_PARASKEUI",
        ora: "_16me17",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "ΑΜΦ",
        didaskon: "Σιάκα Κέρστιν"
    }
]
}, {
    "id" = 4201,
    "title" = "Αντικειμενοστρεφής Προγραμματισμός",
    "eksamino" = "B",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4202,
    "title" = "Εισαγωγή στα Λειτουργικά Συστήματα",
    "eksamino" = "B",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4203,
    "title" = "Διακριτά Μαθηματικά",
    "eksamino" = "B",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4204,
    "title" = "Γλώσσες και Τεχνολογίες Ιστού",
    "eksamino" = "B",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4205,
    "title" = "Πληροφοριακά Συστήματα Ι",
    "eksamino" = "B",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4301,
    "title" = "Αριθμητική Ανάλυση & Προγραμματισμός Επιστημονικών Εφαρμογών",
    "eksamino" = "Γ",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "E",
            tmima: "T1",
            aithousa: "201",
            didaskon: "Αντωνίου Στάθης"
        },
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "E",
            tmima: "T2",
            aithousa: "211",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_DEUTERA",
            ora: "_12me13",
            tipos: "E",
            tmima: "T1",
            aithousa: "201",
        }
    ]
}

```

```

    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_DEUTERA",
    ora: "_12me13",
    tipos: "E",
    tmima: "T2",
    aithousa: "211",
    didaskon: "Ράπτης Πασχάλης"
  },
  {
    mera: "_TRITII",
    ora: "_09me10",
    tipos: "E",
    tmima: "T3",
    aithousa: "211",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_10me11",
    tipos: "E",
    tmima: "T3",
    aithousa: "211",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_11me12",
    tipos: "E",
    tmima: "T4",
    aithousa: "211",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_12me13",
    tipos: "E",
    tmima: "T4",
    aithousa: "211",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_14me15",
    tipos: "0",
    tmima: "01",
    aithousa: "101",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_15me16",
    tipos: "0",
    tmima: "01",
    aithousa: "101",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TRITII",
    ora: "_16me17",
    tipos: "0",
    tmima: "01",
    aithousa: "101",
    didaskon: "Αντωνίου Στάθης"
  },
  {
    mera: "_TETARTI",
    ora: "_16me17",
    tipos: "E",
    tmima: "T5",
    aithousa: "211",
    didaskon: "Αντωνίου Στάθης"
  },

```

```

    {
      mera: "_TETARTI",
      ora: "_17me18",
      tipos: "E",
      tmima: "T5",
      aithousa: "211",
      didaskon: "Αντωνίου Στάθης"
    },
    {
      mera: "_PEMPTI",
      ora: "_09me10",
      tipos: "E",
      tmima: "T6",
      aithousa: "211",
      didaskon: "Ράπτης Πασχάλης"
    },
    {
      mera: "_PEMPTI",
      ora: "_10me11",
      tipos: "E",
      tmima: "T6",
      aithousa: "211",
      didaskon: "Ράπτης Πασχάλης"
    }
  ],
  {
    "id" = 4302,
    "title" = "Δομές Δεδομένων και Ανάλυση Αλγορίθμων",
    "eksamino" = "Γ",
    sisxetiseisOraMathima = [
      {
        mera: "_TRITI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T1",
        aithousa: "202",
        didaskon: "Σταμάτης Δημοσθένης"
      },
      {
        mera: "_TRITI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T1",
        aithousa: "202",
        didaskon: "Σταμάτης Δημοσθένης"
      },
      {
        mera: "_TRITI",
        ora: "_11me12",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "101",
        didaskon: "Σταμάτης Δημοσθένης"
      },
      {
        mera: "_TRITI",
        ora: "_12me13",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "101",
        didaskon: "Σταμάτης Δημοσθένης"
      },
      {
        mera: "_TETARTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T2",
        aithousa: "201",
        didaskon: "Αδαμίδης Παναγιώτης"
      },
      {
        mera: "_TETARTI",
        ora: "_11me12",
        tipos: "E",

```

```

tmima: "T6",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_TETARTI",
ora: "_14me15",
tipos: "E",
tmima: "T3",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_TETARTI",
ora: "_15me16",
tipos: "E",
tmima: "T3",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_PEMPTI",
ora: "_09me10",
tipos: "E",
tmima: "T4",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_PEMPTI",
ora: "_10me11",
tipos: "E",
tmima: "T4",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_PEMPTI",
ora: "_11me12",
tipos: "Θ",
tmima: "Θ1",
aithousa: "ΑΜΦ",
didaskon: "Σταμάτης Δημοσθένης"
},
{
mera: "_PEMPTI",
ora: "_12me13",
tipos: "Θ",
tmima: "Θ1",
aithousa: "ΑΜΦ",
didaskon: "Σταμάτης Δημοσθένης"
},
{
mera: "_PARASKEUI",
ora: "_11me12",
tipos: "E",
tmima: "T5",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_PARASKEUI",
ora: "_12me13",
tipos: "E",
tmima: "T5",
aithousa: "201",
didaskon: "Αδαμίδης Παναγιώτης"
},
{
mera: "_PARASKEUI",
ora: "_14me15",
tipos: "E",
tmima: "T6",
aithousa: "201",

```

```

        didaskon: "Αδαμίδης Παναγιώτης"
    },
    {
        mera: "_PARASKEUI",
        ora: "_15me16",
        tipos: "E",
        tmima: "T6",
        aithousa: "201",
        didaskon: "Αδαμίδης Παναγιώτης"
    }
]
},{
    "id" = 4303,
    "title" = "Οργάνωση και Αρχιτεκτονική Υπολογιστικών Συστημάτων",
    "eksamino" = "Γ",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_DEUTERA",
            ora: "_12me13",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_DEUTERA",
            ora: "_13me14",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T1",
            aithousa: "201",
            didaskon: "Αμανατιάδης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "E",
            tmima: "T1",
            aithousa: "201",
            didaskon: "Αμανατιάδης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_14me15",
            tipos: "E",
            tmima: "T2",
            aithousa: "201",
            didaskon: "Αμανατιάδης Δημήτριος"
        },
        {
            mera: "_TRITI",
            ora: "_15me16",
            tipos: "E",
            tmima: "T2",
            aithousa: "201",
            didaskon: "Αμανατιάδης Δημήτριος"
        },
        {
            mera: "_TETARTI",

```

```

        ora: "_08me09",
        tipos: "E",
        tmima: "T3",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_TETARTI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T3",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_TETARTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_TETARTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_PEMPTI",
        ora: "_14me15",
        tipos: "E",
        tmima: "T5",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_PEMPTI",
        ora: "_15me16",
        tipos: "E",
        tmima: "T5",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_PEMPTI",
        ora: "_16me17",
        tipos: "E",
        tmima: "T6",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
        mera: "_PEMPTI",
        ora: "_17me18",
        tipos: "E",
        tmima: "T6",
        aithousa: "201",
        didaskon: "Αμανατιάδης Δημήτριος"
    }
}
],{
    "id" = 4304,
    "title" = "Αλληλεπίδραση Ανθρώπου-Μηχανής & Ανάπτυξη Διεπιφανειών Χρήστη",
    "eksamino" = "Γ",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "E",
            tmima: "T1",
            aithousa: "202",

```



```

didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_DEUTERA",
  ora: "_12me13",
  tipos: "E",
  tmima: "T1",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_TETARTI",
  ora: "_14me15",
  tipos: "E",
  tmima: "T2",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_TETARTI",
  ora: "_15me16",
  tipos: "E",
  tmima: "T2",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PEMPTI",
  ora: "_14me15",
  tipos: "E",
  tmima: "T3",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PEMPTI",
  ora: "_15me16",
  tipos: "E",
  tmima: "T3",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PEMPTI",
  ora: "_16me17",
  tipos: "E",
  tmima: "T4",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PEMPTI",
  ora: "_17me18",
  tipos: "E",
  tmima: "T4",
  aithousa: "202",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PARASKEUI",
  ora: "_10me11",
  tipos: "0",
  tmima: "01",
  aithousa: "109",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
  mera: "_PARASKEUI",
  ora: "_11me12",
  tipos: "0",
  tmima: "01",
  aithousa: "109",
  didaskon: "Κεραμόπουλος Ευκλείδης"
},

```

```

    {
      mera: "_PARASKEUI",
      ora: "_12me13",
      tipos: "Θ",
      tmima: "Θ1",
      aithousa: "109",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    },
    {
      mera: "_PARASKEUI",
      ora: "_14me15",
      tipos: "Ε",
      tmima: "Τ5",
      aithousa: "202",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    },
    {
      mera: "_PARASKEUI",
      ora: "_15me16",
      tipos: "Ε",
      tmima: "Τ5",
      aithousa: "202",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    }
  ]
}, {
  "id" = 4305,
  "title" = "Συστήματα Διαχείρισης Βάσεων Δεδομένων",
  "eksamino" = "Γ",
  sisxetiseisOraMathima = [
    {
      mera: "_DEUTERA",
      ora: "_09me10",
      tipos: "Ε",
      tmima: "Τ1",
      aithousa: "208",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    },
    {
      mera: "_DEUTERA",
      ora: "_10me11",
      tipos: "Ε",
      tmima: "Τ1",
      aithousa: "208",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    },
    {
      mera: "_DEUTERA",
      ora: "_11me12",
      tipos: "Ε",
      tmima: "Τ2",
      aithousa: "208",
      didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
      mera: "_DEUTERA",
      ora: "_12me13",
      tipos: "Ε",
      tmima: "Τ2",
      aithousa: "208",
      didaskon: "Αμανατιάδης Δημήτριος"
    },
    {
      mera: "_DEUTERA",
      ora: "_14me15",
      tipos: "Ε",
      tmima: "Τ3",
      aithousa: "208",
      didaskon: "Κεραμόπουλος Ευκλείδης"
    },
    {
      mera: "_DEUTERA",
      ora: "_15me16",
      tipos: "Ε",
    }
  ]
}

```

```

tmima: "T3",
aithousa: "208",
didaskon: "Κεραμόπουλος Ευκλείδης"
},
{
mera: "_TRITI",
ora: "_14me15",
tipos: "E",
tmima: "T4",
aithousa: "208",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TRITI",
ora: "_15me16",
tipos: "E",
tmima: "T4",
aithousa: "208",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TRITI",
ora: "_16me17",
tipos: "E",
tmima: "T5",
aithousa: "208",
didaskon: "Αμανατιάδης Δημήτριος"
},
{
mera: "_TRITI",
ora: "_17me18",
tipos: "E",
tmima: "T5",
aithousa: "208",
didaskon: "Αμανατιάδης Δημήτριος"
},
{
mera: "_TETARTI",
ora: "_09me10",
tipos: "Θ",
tmima: "Θ1",
aithousa: "ΑΜΦ",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TETARTI",
ora: "_10me11",
tipos: "Θ",
tmima: "Θ1",
aithousa: "ΑΜΦ",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TETARTI",
ora: "_11me12",
tipos: "Θ",
tmima: "Θ1",
aithousa: "ΑΜΦ",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TETARTI",
ora: "_14me15",
tipos: "E",
tmima: "T6",
aithousa: "208",
didaskon: "Δέρβος Δημήτριος"
},
{
mera: "_TETARTI",
ora: "_15me16",
tipos: "E",
tmima: "T6",
aithousa: "208",

```

```

        didaskon: "Δέρβος Δημήτριος"
    }
}
}, {
    "id" = 4401,
    "title" = "Μεθοδολογίες Προγραμματισμού",
    "eksamino" = "Δ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4402,
    "title" = "Τεχνητή Νοημοσύνη (Γλώσσες & Τεχνικές)",
    "eksamino" = "Δ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4403,
    "title" = "Τηλεπικοινωνίες και Δίκτυα Υπολογιστών",
    "eksamino" = "Δ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4404,
    "title" = "Θεωρία Λειτουργικών Συστημάτων",
    "eksamino" = "Δ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4405,
    "title" = "Θεωρία Πιθανοτήτων & Στατιστική",
    "eksamino" = "Δ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4501,
    "title" = "Πληροφοριακά Συστήματα ΙΙ",
    "eksamino" = "Ε",
    sisxetiseisOraMathima = [
        {
            mera: "_TETARTI",
            ora: "_10me11",
            tipos: "Θ",
            tmima: "01",
            aithousa: "101",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_TETARTI",
            ora: "_11me12",
            tipos: "Θ",
            tmima: "01",
            aithousa: "101",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_PARASKEUI",
            ora: "_11me12",
            tipos: "Θ",
            tmima: "01",
            aithousa: "ΑΜΦ",
            didaskon: "Σιάκα Κέρστιν"
        },
        {
            mera: "_PARASKEUI",
            ora: "_12me13",
            tipos: "Θ",
            tmima: "01",
            aithousa: "ΑΜΦ",
            didaskon: "Σιάκα Κέρστιν"
        }
    ]
}

```

```

    },
    {
        mera: "_PARASKEUI",
        ora: "_13me14",
        tipos: "Θ",
        tmima: "Θ1",
        aithousa: "ΑΜΦ",
        didaskon: "Σιάκα Κέρστιν"
    }
]
},{
    "id" = 4502,
    "title" = "Μηχανική Λογισμικού Ι",
    "eksamino" = "Ε",
    sisxetiseisOraMathima = [
        {
            mera: "_TRITI",
            ora: "_11me12",
            tipos: "Ε",
            tmima: "Τ1",
            aithousa: "210",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TRITI",
            ora: "_12me13",
            tipos: "Ε",
            tmima: "Τ1",
            aithousa: "210",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TRITI",
            ora: "_14me15",
            tipos: "Θ",
            tmima: "Θ1",
            aithousa: "109",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TRITI",
            ora: "_15me16",
            tipos: "Θ",
            tmima: "Θ1",
            aithousa: "109",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TRITI",
            ora: "_16me17",
            tipos: "Θ",
            tmima: "Θ1",
            aithousa: "109",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TETARTI",
            ora: "_14me15",
            tipos: "Ε",
            tmima: "Τ2",
            aithousa: "210",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TETARTI",
            ora: "_15me16",
            tipos: "Ε",
            tmima: "Τ2",
            aithousa: "210",
            didaskon: "Δεληγιάννης Ιγνάτιος"
        },
        {
            mera: "_TETARTI",
            ora: "_16me17",

```

```

        tipos: "E",
        tmima: "T3",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    },
    {
        mera: "_TETARTI",
        ora: "_17me18",
        tipos: "E",
        tmima: "T3",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    },
    {
        mera: "_PEMPTI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T4",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    },
    {
        mera: "_PEMPTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T4",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    },
    {
        mera: "_PEMPTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T5",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    },
    {
        mera: "_PEMPTI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T5",
        aithousa: "210",
        didaskon: "Δεληγιάννης Ιγνάτιος"
    }
]
}, {
    "id" = 4503,
    "title" = "Δίκτυα Υπολογιστών",
    "eksamino" = "E",
    sisxetiseisOraMathima = [
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T1",
            aithousa: "208",
            didaskon: "Ψαρράς Νικόλαος"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "E",
            tmima: "T1",
            aithousa: "208",
            didaskon: "Ψαρράς Νικόλαος"
        },
        {
            mera: "_TRITI",
            ora: "_11me12",
            tipos: "E",
            tmima: "T2",
            aithousa: "208",
            didaskon: "Ψαρράς Νικόλαος"
        }
    ]
}

```

```

    },
    {
        mera: "_TRITI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T2",
        aithousa: "208",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T3",
        aithousa: "208",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T3",
        aithousa: "208",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T4",
        aithousa: "208",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_14me15",
        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_15me16",
        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Ψαρράς Νικόλαος"
    },
    {
        mera: "_PEMPTI",
        ora: "_16me17",
        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Ψαρράς Νικόλαος"
    }
}
],
{
    "id" = 4504,
    "title" = "Ανάπτυξη Διαδικτυακών Συστημάτων & Εφαρμογών",
    "eksamino" = "E",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_14me15",

```

```

        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    },
    {
        mera: "_DEUTERA",
        ora: "_15me16",
        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    },
    {
        mera: "_TRITI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T1",
        aithousa: "201",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_TRITI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T1",
        aithousa: "201",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_PEMPTI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T2",
        aithousa: "202",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_PEMPTI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T2",
        aithousa: "202",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_PEMPTI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_PEMPTI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Διαμαντάρας Κωνσταντίνος"
    },
    {
        mera: "_PARASKEUI",
        ora: "_14me15",
        tipos: "0",
        tmima: "01",
        aithousa: "109",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    },
    {
        mera: "_PARASKEUI",
        ora: "_15me16",
        tipos: "0",
        tmima: "01",
    }

```



```

        aithousa: "109",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    }
]
}, {
    "id" = 4505,
    "title" = "Επιχειρησιακή Έρευνα",
    "eksamino" = "Ε",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "101",
            didaskon: "Κώστογλου Βασίλειος"
        },
        {
            mera: "_DEUTERA",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "101",
            didaskon: "Κώστογλου Βασίλειος"
        },
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",
            aithousa: "101",
            didaskon: "Κώστογλου Βασίλειος"
        },
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "101",
            didaskon: "Κώστογλου Βασίλειος"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "101",
            didaskon: "Κώστογλου Βασίλειος"
        }
    ]
}, {
    "id" = 4601,
    "title" = "Ασφάλεια Πληροφοριακών Συστημάτων",
    "eksamino" = "ΣΤ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4602,
    "title" = "Μηχανική Μάθηση",
    "eksamino" = "ΣΤ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4603,
    "title" = "Τεχνολογία Βάσεων Δεδομένων",
    "eksamino" = "ΣΤ",
    sisxetiseisOraMathima = [
    ]
}, {
    "id" = 4604,
    "title" = "Μηχανική Λογισμικού ΙΙ",

```

```

"eksamino" = "ΣΤ",
sisxetiseisOraMathima = [
]
},{
"id" = 4701,
"title" = "Ανάπτυξη και Διαχείριση Ολοκληρωμένων Πλ. Συστημάτων & Εφαρμογών",
"eksamino" = "Ζ",
sisxetiseisOraMathima = [
  {
    mera: "_DEUTERA",
    ora: "_11me12",
    tipos: "E",
    tmima: "T1",
    aithousa: "301",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_DEUTERA",
    ora: "_12me13",
    tipos: "E",
    tmima: "T1",
    aithousa: "301",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_TRITI",
    ora: "_11me12",
    tipos: "E",
    tmima: "T2",
    aithousa: "301",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_TRITI",
    ora: "_12me13",
    tipos: "E",
    tmima: "T2",
    aithousa: "301",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_TRITI",
    ora: "_14me15",
    tipos: "Θ",
    tmima: "Θ1",
    aithousa: "102",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_TRITI",
    ora: "_15me16",
    tipos: "Θ",
    tmima: "Θ1",
    aithousa: "102",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_PEMPTI",
    ora: "_14me15",
    tipos: "Θ",
    tmima: "Θ1",
    aithousa: "102",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  },
  {
    mera: "_PEMPTI",
    ora: "_15me16",
    tipos: "Θ",
    tmima: "Θ1",
    aithousa: "102",
    didaskon: "Σαλαμπάσης Μιχαήλ"
  }
]
}

```

```

        mera: "_PARASKEUI",
        ora: "_11me12",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    },
    {
        mera: "_PARASKEUI",
        ora: "_12me13",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Σαλαμπάσης Μιχαήλ"
    }
]
},{
    "id" = 4702,
    "title" = "Τεχνολογία Πολυμέσων",
    "eksamino" = "Z",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_09me10",
            tipos: "E",
            tmima: "T1",
            aithousa: "301",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_DEUTERA",
            ora: "_10me11",
            tipos: "E",
            tmima: "T1",
            aithousa: "301",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "E",
            tmima: "T2",
            aithousa: "301",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "E",
            tmima: "T2",
            aithousa: "301",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_TETARTI",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_TETARTI",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "109",
            didaskon: "Κλεφτούρης Δημήτριος"
        },
        {
            mera: "_TETARTI",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",

```

```

        aithousa: "109",
        didaskon: "Κλεφτούρης Δημήτριος"
    },
    {
        mera: "_PARASKEUI",
        ora: "_09me10",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Ράπτης Πασχάλης"
    },
    {
        mera: "_PARASKEUI",
        ora: "_10me11",
        tipos: "E",
        tmima: "T3",
        aithousa: "301",
        didaskon: "Ράπτης Πασχάλης"
    }
]
}, {
    "id" = 4801,
    "title" = "Ευφυή Συστήματα",
    "eksamino" = "Z",
    sisxetiseisOraMathima = [
]
}, {
    "id" = 4802,
    "title" = "Προηγμένες Αρχιτεκτονικές Υπολογιστών και Παράλληλα Συστήματα",
    "eksamino" = "Z",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Διαμαντάρας Κωνσταντίνος"
        },
        {
            mera: "_DEUTERA",
            ora: "_12me13",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Διαμαντάρας Κωνσταντίνος"
        },
        {
            mera: "_DEUTERA",
            ora: "_13me14",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Διαμαντάρας Κωνσταντίνος"
        },
        {
            mera: "_TRITI",
            ora: "_09me10",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Διαμαντάρας Κωνσταντίνος"
        },
        {
            mera: "_TRITI",
            ora: "_10me11",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Διαμαντάρας Κωνσταντίνος"
        }
    ]
}, {

```

```

"id" = 4803,
"title" = "Οργάνωση Δεδομένων και Εξόρυξη Πληροφορίας",
"eksamino" = "Z",
sisxetiseisOraMathima = [
  {
    mera: "_TRITI",
    ora: "_11me12",
    tipos: "0",
    tmima: "01",
    aithousa: "102",
    didaskon: "Κεραμόπουλος Ευκλείδης"
  },
  {
    mera: "_TRITI",
    ora: "_12me13",
    tipos: "0",
    tmima: "01",
    aithousa: "102",
    didaskon: "Κεραμόπουλος Ευκλείδης"
  },
  {
    mera: "_TRITI",
    ora: "_13me14",
    tipos: "0",
    tmima: "01",
    aithousa: "102",
    didaskon: "Κεραμόπουλος Ευκλείδης"
  },
  {
    mera: "_PEMPTI",
    ora: "_09me10",
    tipos: "0",
    tmima: "01",
    aithousa: "102",
    didaskon: "Δέρβος Δημήτριος"
  },
  {
    mera: "_PEMPTI",
    ora: "_10me11",
    tipos: "0",
    tmima: "01",
    aithousa: "102",
    didaskon: "Δέρβος Δημήτριος"
  }
],{
  "id" = 4804,
  "title" = "Ειδ. Θέματα Δικτύων Ι",
  "eksamino" = "Z",
  sisxetiseisOraMathima = [
    {
      mera: "_DEUTERA",
      ora: "_08me09",
      tipos: "0",
      tmima: "01",
      aithousa: "210",
      didaskon: "Ψαρράς Νικόλαος"
    },
    {
      mera: "_DEUTERA",
      ora: "_09me10",
      tipos: "0",
      tmima: "01",
      aithousa: "210",
      didaskon: "Ψαρράς Νικόλαος"
    },
    {
      mera: "_DEUTERA",
      ora: "_10me11",
      tipos: "E",
      tmima: "T1",
      aithousa: "210",
      didaskon: "Ψαρράς Νικόλαος"
    }
  ],

```

```

        {
            mera: "_DEUTERA",
            ora: "_11me12",
            tipos: "E",
            tmima: "T1",
            aithousa: "210",
            didaskon: "Ψαρράς Νικόλαος"
        },
        {
            mera: "_DEUTERA",
            ora: "_12me13",
            tipos: "E",
            tmima: "T1",
            aithousa: "210",
            didaskon: "Ψαρράς Νικόλαος"
        },
        {
            mera: "_DEUTERA",
            ora: "_13me14",
            tipos: "E",
            tmima: "T1",
            aithousa: "210",
            didaskon: "Ψαρράς Νικόλαος"
        }
    ],
    },{
        "id" = 4805,
        "title" = "Ειδ. Θέματα Δικτύων II",
        "eksamino" = "Z",
        sisxetiseisOraMathima = [
            ]
    },{
        "id" = 4806,
        "title" = "Διαδικτυακές Υπηρεσίες Προστιθέμενης Αξίας",
        "eksamino" = "Z",
        sisxetiseisOraMathima = [
            {
                mera: "_DEUTERA",
                ora: "_09me10",
                tipos: "Θ",
                tmima: "Θ1",
                aithousa: "102",
                didaskon: "Ηλιούδης Χρήστος"
            },
            {
                mera: "_DEUTERA",
                ora: "_10me11",
                tipos: "Θ",
                tmima: "Θ1",
                aithousa: "102",
                didaskon: "Ηλιούδης Χρήστος"
            },
            {
                mera: "_TETARTI",
                ora: "_14me15",
                tipos: "Θ",
                tmima: "Θ1",
                aithousa: "109",
                didaskon: "Ηλιούδης Χρήστος"
            },
            {
                mera: "_PEMPTI",
                ora: "_11me12",
                tipos: "Θ",
                tmima: "Θ1",
                aithousa: "102",
                didaskon: "Ηλιούδης Χρήστος"
            },
            {
                mera: "_PEMPTI",
                ora: "_12me13",
                tipos: "Θ",
                tmima: "Θ1",
            }
        ]
    }

```

```

        aithousa: "102",
        didaskon: "Ηλιούδης Χρήστος"
    }
]
},{
    "id" = 4807,
    "title" = "Ασύρματα και Κινητά Δίκτυα Επικοινωνιών",
    "eksamino" = "Z",
    sisxetiseisOraMathima = [
]
},{
    "id" = 4808,
    "title" = "Γραφικά Υπολογιστών",
    "eksamino" = "Z",
    sisxetiseisOraMathima = [
        {
            mera: "_DEUTERA",
            ora: "_14me15",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_DEUTERA",
            ora: "_15me16",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_DEUTERA",
            ora: "_16me17",
            tipos: "0",
            tmima: "01",
            aithousa: "102",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_PARASKEUI",
            ora: "_11me12",
            tipos: "0",
            tmima: "01",
            aithousa: "208",
            didaskon: "Ράπτης Πασχάλης"
        },
        {
            mera: "_PARASKEUI",
            ora: "_12me13",
            tipos: "0",
            tmima: "01",
            aithousa: "208",
            didaskon: "Ράπτης Πασχάλης"
        }
    ]
}
]

```

Παράρτημα 2: Κώδικας για το Action Bar

XML κώδικας για το Action Bar:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp"
    android:background="@color/TITLEBAR"
    >

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="fill_parent"
        android:layout_gravity="center_horizontal"
        android:src="@drawable/ic_launcher" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:layout_gravity="center"
        android:layout_marginRight="5dp"
        android:layout_weight="1"
        android:textColor="#FFFFFF"
        android:shadowColor="#333333"
        android:shadowDx="1"
        android:shadowDy="1"
        android:shadowRadius="2"
        android:id="@+id/titleTvLeft"
        android:text="left">
    </TextView>

    <ProgressBar
        android:id="@+id/LeadProgressBar"
        style="?android:attr/progressBarStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="left"
        android:layout_gravity="center"
        android:layout_marginRight="5dp">
    </ProgressBar>

    <Button
        android:background="@android:drawable/ic_menu_more"
        android:drawableLeft="@android:drawable/ic_menu_more"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/titleBtn"
    />

</LinearLayout>
```


JAVA κώδικας για το Action Bar:

```
[...]
//set custom titlebar
KatyMethods.customTitleBar(this, getText(R.string.title_activity_edit_modules).toString(), "save");

// show the progress bar because loading data will last
titleProgressBar = (ProgressBar) findViewById(R.id.LeadProgressBar);
titleProgressBar.setVisibility(ProgressBar.VISIBLE);

[...]

public void customTitleBar(String left, String right) {

    if (right.length() > 20)
        right = right.substring(0, 20);

    //set up custom title
    TextView titleTvLeft = (TextView) findViewById(R.id.titleTvLeft);
    titleTvLeft.setText(left);

    titleProgressBar = (ProgressBar) findViewById(R.id.LeadProgressBar);
}
```