



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Προσφορά και ζήτηση αγοράς εργασίας σε λειτουργικό Android



Του φοιτητή
Σάββου Βαγγέλη
Αρ. Μητρώου: 08/3385

Επιβλέπον καθηγητής
Ευκλείδης Κεραμόπουλος

Θεσσαλονίκη 2014

ΠΡΟΛΟΓΟΣ

Τα τελευταία χρόνια είμαστε μάρτυρες μιας τεχνολογικής επανάστασης. Οι τεχνολογίες των πληροφοριών και των επικοινωνιών (ΤΠΕ) αποτελούν ισχυρό κινητήρα ανάπτυξης και απασχόλησης. Οι υπηρεσίες ΤΠΕ και οι σχετικές δεξιότητες, τα μέσα επικοινωνίας και το περιεχόμενο συνιστούν αυξανόμενο μερίδιο της οικονομίας και της κοινωνίας.

Ο ηλεκτρονικός υπολογιστής και οι κινητές συσκευές (Κινητά, PDAs, laptops) ως εργαλείο εργασίας, αλλά και ψυχαγωγίας – ενημέρωσης – επικοινωνίας, δεν αφορούν πλέον μόνο τα εξειδικευμένα επαγγέλματα και τον επιστημονικό τομέα, αλλά έχουν διεισδύσει σε όλους τους τομείς εργασίας, αποτελώντας πλέον αναπόσπαστο κομμάτι της καθημερινότητας σχεδόν όλων των ατόμων.

Όλο και περισσότερες χρήσεις και λειτουργίες του Η/Υ αντικαθιστούνται με την χρήση των κινητών τηλεφώνων, πράγμα που σημαίνει ότι πέρα από την λαμπρή ανάπτυξη της τεχνολογίας τα τελευταία χρόνια έχουμε ακόμη μία τερατώδη ανάπτυξη στην παραγωγή και λειτουργικότητα κινητών τηλεφώνων καθώς οι χρήστες επιζητούν ολοένα και περισσότερες λειτουργίες και περισσότερη ενέργεια σε μία συσκευή μερικών εκατοστών για να ολοκληρώνουν τις δουλειές τους γρήγορα και σε οποιοδήποτε σημείο του πλανήτη.

Για αυτό το λόγω και εμείς σαν αεραγωγοί της τεχνολογίας και σαν προγραμματιστές θα επιτελέσουμε σε αυτό το έργο με γνώμονα τον χρήστη και τις ανάγκες του για την υλοποίηση αυτής της εφαρμογής Android.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως κύριο στόχο την ανάπτυξη μίας εφαρμογής σε περιβάλλον Android. Η εφαρμογή επιτρέπει στον κάτοχο κινητού τηλεφώνου με το λειτουργικό σύστημα Android να βρίσκει αλλά και να προσφέρει αγορά εργασίας.

Τα βασικά στοιχεία αυτής της πτυχιακής εργασίας επικεντρώνονται σε δύο βασικούς άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους και την αισθητική παρουσίαση της εφαρμογής, και οριοθετεί την ευχρηστία, την ευκολία πρόσβασης και την “ελαφρότητά” της μέσω του interface ενός “έξυπνου κινητού τηλεφώνου” (smart phone). Ο δεύτερος αφορά την αξιοπιστία και τη διασύνδεση με την εξωτερική βάση δεδομένων καθώς και τεχνολογίες που χρησιμοποιήθηκαν προκειμένου να επιτευχθεί το μέγιστο δυνατό αποτέλεσμα.

Συμπερασματικά, λοιπόν, πρόκειται για μία ολοκληρωμένη εφαρμογή σε περιβάλλον Android, η οποία αναμένεται να τύχει ευρείας αποδοχής από τους χρήστες έξυπνων κινητών τηλεφώνων και η οποία είναι ιδιαίτερα εύπλαστη και σε θέση να ακολουθεί κάθε φορά τις εκάστοτε παρουσιαζόμενες ανάγκες διαχείρισης πολυμεσικού περιεχομένου από τους ίδιους τους χρήστες (user-generated content).

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία πραγματοποιήθηκε στο Αλεξάνδρειο τεχνολογικό εκπαιδευτικό ίδρυμα, στο τμήμα πληροφορικής κατά το έτος 2014.

Η ολοκλήρωση της πτυχιακής αυτής εργασίας θα ήταν αδύνατη χωρίς την πολύτιμη υποστήριξη του καθηγητή μου, κ. Κεραμόπουλου Ευκλείδη. Χρωστάω επίσης, ένα μεγάλο ευχαριστώ στους καθηγητές του τμήματος πληροφορικής που μου προσέφεραν όλα αυτά τα χρόνια τις πολύτιμες γνώσεις τους έτσι ώστε να βγω στην αγορά εργασίας με τα καλύτερα εφόδια.

Ευχαριστώ πολύ τους καλούς μου φίλους, Χριστόφορο Μαγγουλίδη, Χρήστο Τσολάκη, Ηλέκτρα Δελή, Δημήτρη Αμουτζήδη για την συνεχή συμπαράσταση και επικοινωνία που είχαμε όλο αυτό το διάστημα.

Τέλος, θέλω να ευχαριστήσω πολύ τους γονείς μου, Σάββο Μανώλη και Σάββου Ελευθερία, οι οποίοι υπήρξαν πάντα ένα ανεκτίμητο στήριγμα για μένα και στους οποίους οφείλω όλη τη διαδρομή των σπουδών μου, μέχρι σήμερα.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	2
ΠΕΡΙΛΗΨΗ	3
ΕΥΧΑΡΙΣΤΙΕΣ	4
Περιεχόμενα.....	5
Εισαγωγή	8
1 Εισαγωγή στο λειτουργικό σύστημα Android	9
1.1 Εισαγωγή	9
1.2 Τί είναι το Android	9
1.3 Ιστορική Αναδρομή	9
1.4 Κύρια χαρακτηριστικά.....	11
1.4.1 Ανάπτυξη ισχυρών εφαρμογών	11
1.4.2 Πλούσια και ασφαλής ενοποίηση εφαρμογών.....	11
1.4.3 Κανένα δαπανηρό εμπόδιο για τη δημοσίευση	12
1.4.4 Δωρεάν και ανοιχτή Πηγή	12
1.4.5 Οικεία και οικονομικά εργαλεία ανάπτυξης.....	13
1.4.6 Δωρεάν διαθέσιμα εργαλεία ανάπτυξης λογισμικού	13
1.4.7 Οικεία γλώσσα, οικεία περιβάλλοντα ανάπτυξης.....	13
1.4.8 Google Play Store	13
1.5 Αρχιτεκτονική.....	14
1.6 Οι εκδόσεις του Android.....	16
1.7 Android vs. iOS	20
1.8 Επίλογος.....	21
2 Προγραμματισμός και ανάπτυξη εφαρμογών σε περιβάλλον Android.....	22
2.1 Εισαγωγή	22
2.2 Κατηγορίες Εφαρμογών.....	22
2.2.1 Εφαρμογές προσκηνίου	22
2.2.2 Εφαρμογές παρασκηνίου	23
2.2.3 Διακοπτόμενες εφαρμογές.....	23
2.2.4 Widgets & Live Wallpapers.....	23
2.3 Συστατικά στοιχεία μιας εφαρμογής.....	24
2.3.1 Context (Περιβάλλον).....	24
2.3.2 Activity (Δραστηριότητα).....	24
2.3.3 Intent (Πρόθεση).....	27
2.3.4 Service (Υπηρεσία):.....	27
2.3.5 Fragments.....	27
2.3.6 Broadcast Receivers.....	30
2.3.7 Processes and Threads (Διαδικασίες και Νήματα)	30
2.3.7.1 GUI Thread	30
2.3.7.2 AsyncTask.....	31
2.3.8 User Interface.....	31
2.3.8.1 Layouts.....	32
2.3.8.1.1 FrameLayout	32
2.3.8.1.2 LinearLayout.....	33
2.3.8.1.3 RelativeLayout.....	34

2.3.8.1.4	TableLayout	36
2.3.8.2	ListView.....	37
2.3.8.3	Dialogs	38
2.3.8.4	Action Bar.....	39
2.3.8.5	Input Control.....	40
2.4	Επίλογος.....	40
3	Αρχιτεκτονική και τεχνολογίες που χρησιμοποιήθηκαν	41
3.1	Εισαγωγή	41
3.2	Περιγραφή του Project.....	41
3.3	Δομή του project στο Eclipse	42
3.4	Τεχνολογίες και γλώσσες προγραμματισμού που χρησιμοποιήθηκαν	48
3.4.1	Γραφικός σχεδιασμός εφαρμογής.....	49
3.4.2	Γλώσσες προγραμματισμού.....	50
3.4.2.1	Java	50
3.4.2.2	PHP.....	51
3.4.2.2.1	Γιατί PHP	52
3.4.2.3	XML.....	52
3.4.2.3.1	Βασική ορολογία.....	53
3.4.2.3.2	Παράδειγμα.....	55
3.4.2.4	MySQL	55
3.4.2.4.1	Χαρακτηριστικά-Πλεονεκτήματα.....	56
3.4.3	Προγραμματιστικά περιβάλλοντα	58
3.4.3.1	Eclipse.....	58
3.4.3.1.1	Προσαρμογή του Android στο eclipse.....	59
3.4.3.2	Aptana.....	60
3.4.3.3	Filezilla.....	61
3.4.3.4	PhpMyAdmin.....	62
3.4.3.4.1	Χαρακτηριστικά.....	63
3.4.4	Τεχνολογίες που χρησιμοποιήθηκαν	63
3.4.4.1	JSON.....	63
3.4.4.1.1	JSON μορφές	64
3.4.4.1.2	Παράδειγμα.....	67
3.4.4.1.3	Json Parser	68
3.4.4.2	GCM (Google Cloud Message)	68
3.4.4.2.1	Γιατί Google Cloud Message.....	69
3.4.4.2.2	Δομή.....	69
3.4.4.3	Κρυπτογράφηση-Αποκρυπτογράφηση	71
3.4.4.3.1	Στόχοι.....	72
3.4.4.3.2	Διαδικασία κρυπτογράφησης –αποκρυπτογράφησης.....	72
3.5	Επίλογος.....	74
4	Αναλυτική λειτουργία και χρήση της εφαρμογής.....	75
4.1	Εισαγωγή	75
4.2	Αρχική οθόνη.....	75
4.3	Log In Form	76
4.3.1	Forgot your password Form.....	77
4.3.2	Register Form.....	78

4.4	Search Form	79
4.5	CV Form	82
4.6	Job Offer Form.....	83
4.7	Επίλογος.....	84
5	Επίλογος.....	85
5.1	Σύνοψη και συμπεράσματα	85
5.2	Μελλοντικές επεκτάσεις.....	85
6	Βιβλιογραφία	87
	Παράστημα Α	89

Εισαγωγή

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι ο σχεδιασμός και η δημιουργία μίας εφαρμογής με τις πιο σύγχρονες τεχνολογίες που χρησιμοποιούνται αυτήν την στιγμή στην ανάπτυξη Android εφαρμογών.

Στα πρώτα κεφάλαια γίνεται μια ιστορική αναδρομή για τις εκδόσεις και τις δυνατότητες της νέας πλατφόρμας, καθώς και αναλυτική περιγραφή των συστατικών στοιχείων που απαρτίζουν μία εφαρμογή Android.

Στα επόμενα κεφάλαια γίνεται διεξοδική περιγραφή των τεχνολογιών και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν καθώς και μια αναλυτική λειτουργία και χρήση της εφαρμογής, αλλά και περιγραφή των εργαλείων προγραμματισμού που χρειάστηκαν για την δημιουργία της εφαρμογής με βάση το Eclipse.

Η εφαρμογή σχετίζεται με την προσφορά και ζήτηση αγοράς εργασίας με σκοπό την μείωση των αποστάσεων ανάμεσα στον εργοδότη και τον εργαζόμενο αλλά και την ευκολότερη και γρηγορότερη κάλυψη των θέσεων εργασίας με βάση τα προσόντα του καθενός.

Καλή ανάγνωση!

1 Εισαγωγή στο λειτουργικό σύστημα Android

1.1 Εισαγωγή

Σύμφωνα λοιπόν με τα παραπάνω, σε αυτό το πρώτο εισαγωγικό κεφάλαιο επεξηγείται το τι ακριβώς είναι αυτό το λειτουργικό σύστημα πάνω στο οποίο θα αναπτυχθεί η εφαρμογή Android (προσφορά και ζήτηση αγοράς εργασίας). Επιπρόσθετα θα γίνει μία ανάλυση για το πού αλλά και το πώς ξεκίνησε καθώς και την εξέλιξη του στην πορεία των χρόνων. Θα αναλυθεί η αρχιτεκτονική του αλλά το πιο σημαντικό, ποια είναι εκείνα τα χαρακτηριστικά τα οποία του δίνουν την πρωτιά τόσο για τους προγραμματιστές Android εφαρμογών όσο και για τους χρήστες του.

1.2 Τί είναι το Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

1.3 Ιστορική Αναδρομή

Η Google, έχοντας εντοπίσει αυξημένη χρήση του internet και σχετικών αναζητήσεων στον παγκόσμιο ιστό μέσω κινητών συσκευών (mobile devices) εξαγοράζει το 2005 την Android Inc με σκοπό την ανάπτυξη μιας πλατφόρμας για τέτοιου είδους συσκευές. Περίπου την ίδια περίοδο, η Apple παρουσιάζει το iPhone (2007) το οποίο κάνει χρήση κάποιων επαναστατικών καινοτομιών όπως η υποστήριξη multitouch και η ανοιχτή παγκόσμια αγορά εφαρμογών. Το Android γρήγορα προσαρμόστηκε ώστε να υιοθετήσει και να παρέχει επίσης τις προαναφερθείσες δυνατότητες αν και κατά κοινή ομολογία οι πρώτες εκδόσεις του, υπολείπονταν του iPhone όσον αφορά τις υποστηριζόμενες λειτουργίες και τη συνολική εμπειρία που πρόσφερε στους χρήστες.

Το 2007 λοιπόν, τη χρονιά λανσαρίσματος του iPhone, δημιουργείται ένας οργανισμός που αποτελείται από μεγάλο αριθμό εταιρειών τηλεπικοινωνιακού εξοπλισμού καθώς και εταιρείες πληροφορικής όπως η Google, η T-Mobile, η Motorola, η Samsung, η Sony Ericsson, η Intel, η Vodafone, η Toshiba κ.α. με όνομα Open Handset Alliance (<http://www.openhandsetalliance.com>) και σκοπό την έρευνα και την ανάπτυξη τεχνολογιών για την παραγωγή συσκευών που θα διευκολύνουν τόσο τους παρόχους κινητής τηλεφωνίας όσο και τους κατασκευαστές κινητών τηλεφώνων αλλά και τους προγραμματιστές εφαρμογών. Τα μέλη της συμμαχίας δεσμεύτηκαν να παρέχουν τις τεχνολογίες αυτές βάσει του μοντέλου ανοιχτού πηγαίου κώδικα Apache.

Η πρώτη 'early look' έκδοση του Android SDK δημοσιεύτηκε το Νοέμβριο του 2007, ενώ το πρώτο smart phone που έκανε χρήση λειτουργικού Android ήταν το G1 της T-Mobile. Οι συσκευές Android άρχισαν να διαδίδονται με γρήγορο ρυθμό κυρίως λόγω της δυνατότητας της πλατφόρμας να εκμεταλλεύεται το μοντέλο cloud computing αλλά και της έμφυτης υποστήριξης για συνεργασία με μία σχεσιακή βάση δεδομένων (SQLite).

Ακολούθησαν αρκετές αναβαθμισμένες εκδόσεις του Android, κάθε μία προσθέτοντας νέα χαρακτηριστικά και λειτουργίες. Για κάποιον άγνωστο λόγο κάθε έκδοση του Android φέρει και μία κωδική ονομασία ενός γλυκού εδέσματος (1.5 - Cupcake, 1.6 - Donut, 2.0 Eclair κ.α.), ενώ από τις εκδόσεις αυτές εκείνη που εισήγαγε την υποστήριξη πιο ανεπτυγμένων λειτουργιών ήταν σίγουρα η 2.0 στην οποία ενσωματώθηκε η υποστήριξη multitouch, HTML 5, text-to-speech και η δυνατότητα πιο προχωρημένων αναζητήσεων. Πιο συγκεκριμένα :

- ✚ Multitouch: Δίνεται η δυνατότητα στον χρήστη να αλληλεπιδρά με το μηχάνημα παραπάνω από μία φορά την ίδια χρονική στιγμή.
- ✚ HTML 5: Είναι μια γλώσσα σήμανσης για τον Παγκόσμιο Ιστό που είναι η επόμενη μεγάλη έκδοση της HTML(Γλώσσα Υπερκειμένου, HyperText Markup Language).
- ✚ text-to-speech: Σύνθεση φωνής από κείμενο σε διάφορες γλώσσες.

Κατά τη σύνταξη της πτυχιακής αυτής, η πιο πρόσφατη έκδοση του λειτουργικού Android είναι η 4.4.4 (Kit Kat).

1.4 Κύρια χαρακτηριστικά

Τα κύρια χαρακτηριστικά που έκαναν το λειτουργικό αυτό σύστημα που ονομάζεται Android τόσο διαδεδομένο και νούμερο ένα στις πωλήσεις κινητών συσκευών ακολουθούν παρακάτω.

1.4.1 Ανάπτυξη ισχυρών εφαρμογών

Στο παρελθόν, οι κατασκευαστές τηλεφωνικών συσκευών συχνά καθόριζαν ειδικές σχέσεις με αξιόπιστους ανεξάρτητους προγραμματιστές λογισμικού. Αυτή η ομάδα προγραμματιστών λογισμικού έγραφαν εσωτερικές εφαρμογές όπως η ανταλλαγή μηνυμάτων και τα προγράμματα περιήγησης στο web, οι οποίες περιλαμβάνονταν στο βασικό σύνολο χαρακτηριστικών του τηλεφώνου. Για τη σχεδίαση αυτών των εφαρμογών, ο κατασκευαστής παραχωρούσε στον προγραμματιστή δικαίωμα εξουσιοδοτημένης πρόσβασης και γνώσης του εσωτερικού λογισμικού και του firmware της συσκευής. Στην πλατφόρμα Android δεν υπάρχει καμιά διάκριση μεταξύ των εσωτερικών των ανεξάρτητων εφαρμογών, κάτι που επιτρέπει τον υγιή ανταγωνισμό μεταξύ προγραμματιστών εφαρμογών. Όλες οι εφαρμογές Android χρησιμοποιούν τις ίδιες βιβλιοθήκες. Οι εφαρμογές Android διαθέτουν πρωτοφανή δικαιώματα πρόσβασης στο υποκείμενο υλικό, δίνοντας το δικαίωμα στους προγραμματιστές να δημιουργούν όλο και πιο ισχυρές εφαρμογές. Οι εφαρμογές μπορούν να διευρυνθούν ή να αντικατασταθούν.

1.4.2 Πλούσια και ασφαλής ενοποίηση εφαρμογών

Ένα από τα πιο ανταγωνιστικά και καινοτόμα χαρακτηριστικά της πλατφόρμας του Android είναι η καλά σχεδιασμένη ενοποίηση εφαρμογών. Το Android παρέχει όλα τα εργαλεία που είναι απαραίτητα για την κατασκευή μίας καλύτερης εφαρμογής, για παράδειγμα, επιτρέποντας στους προγραμματιστές να γράφουν εφαρμογές οι οποίες εκμεταλλεύονται βασικές λειτουργίες όπως η περιήγηση στο web, η εύρεση στο χάρτη, η διαχείριση επαφών και η αποστολή μηνυμάτων. Οι εφαρμογές μπορούν επίσης να γίνουν πάροχοι υλικού και να μοιράζονται τα δεδομένα μεταξύ τους με ασφαλή τρόπο. Πλατφόρμες όπως το

Symbian υποφέρουν από επιθέσεις κακόβουλου λογισμικού. Το σφριγηλό μοντέλο ασφάλειας εφαρμογών του Android προστατεύει τον χρήστη και το σύστημα από επικίνδυνο λογισμικό.

1.4.3 Κανένα δαπανηρό εμπόδιο για τη δημοσίευση

Οι εφαρμογές Android δεν έχουν καμιά σχέση με τα ακριβά και χρονοβόρα προγράμματα δοκιμών και πιστοποίησης, τα οποία είναι υποχρεωτικά σε άλλες πλατφόρμες.

1.4.4 Δωρεάν και ανοιχτή Πηγή

Το Android είναι μία πλατφόρμα ανοιχτής πηγής. Ούτε οι προγραμματιστές, ούτε οι κατασκευαστές τηλεφωνικών συσκευών δεν υποχρεούνται να καταβάλλουν ποσά για πνευματικά δικαιώματα ή για τη χρήση της πλατφόρμας. Το υποκείμενο λειτουργικό σύστημα του Android έχει κατοχυρωθεί με την άδεια δημόσιας χρήσης GNU General Public License Version 2 (GPLv2), μία ισχυρή άδεια που υποχρεώνει τις βελτιώσεις τρίτων να εξακολουθούν να εμπίπτουν στους όρους των προτύπων ανοιχτής πηγής. Το περιβάλλον δημιουργίας του Android διανέμεται μέσω της άδειας Apache Software License (ASL/ Apache2), η οποία προβλέπει τη διανομή παρεκκλίσεων ανοιχτής και κλειστής πηγής του πηγαίου κώδικα. Οι εμπορικοί προγραμματιστές (ιδίως οι κατασκευαστές τηλεφωνικών συσκευών) μπορούν να επιλέξουν να ενισχύσουν την πλατφόρμα χωρίς να είναι υποχρεωμένοι να παρέχουν τις βελτιώσεις τους στην κοινότητα ανοιχτής πηγής. Αντίθετα, οι προγραμματιστές μπορούν να επωφεληθούν απ' τις βελτιώσεις, που αφορούν σε συγκεκριμένες συσκευές και να αναδιανείμουν το έργο τους με τους όρους οποιασδήποτε άδειας χρήσης θέλουν. Οι προγραμματιστές εφαρμογών Android έχουν τη δυνατότητα να διανέμουν τις εφαρμογές τους υπό οποιαδήποτε άδεια προτιμούν. Οι προγραμματιστές μπορούν να γράφουν δωρεάν λογισμικό ανοιχτής πηγής ή κλασικές εφαρμογές, των οποίων η πνευματική ιδιοκτησία προστατεύεται, για εμπορική εκμετάλλευση, αλλά και με οποιονδήποτε άλλο τρόπο.

1.4.5 Οικεία και οικονομικά εργαλεία ανάπτυξης

Αντίθετα από κάποιες ιδιωτικές πλατφόρμες όπως η πλατφόρμα του IoS που απαιτούν αμοιβές για την εγγραφή των προγραμματιστών (50€/χρόνο), προσεκτική εξέταση και ακριβούς μεταγλωττιστές, δεν υπάρχουν προκαταβολικά κόστη για την ανάπτυξη εφαρμογών Android.

1.4.6 Δωρεάν διαθέσιμα εργαλεία ανάπτυξης λογισμικού

Το SDK και τα εργαλεία του Android διατίθενται δωρεάν. Οι προγραμματιστές μπορούν να λάβουν το Android SDK απ' το website του Android, αφού συμφωνήσουν στους όρους της συμφωνίας άδειας χρήσης του πακέτου ανάπτυξης λογισμικού του Android.

1.4.7 Οικεία γλώσσα, οικεία περιβάλλοντα ανάπτυξης

Οι προγραμματιστές έχουν αρκετές επιλογές όσον αφορά στα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE). Πολλοί προγραμματιστές επιλέγουν το δημοφιλές και δωρεάν Eclipse IDE για τη σχεδίαση και ανάπτυξη εφαρμογών Android.

Οι εφαρμογές Android μπορούν να αναπτυχθούν στα παρακάτω λειτουργικά συστήματα:

- Windows XP (32-bit) ή Vista ή Windows 8(32 bit ή 64 bit)
- Mac OS X 10.5.8 ή επόμενη έκδοση (μόνο για x86)
- Linux (δοκιμασμένο σε Linux Ubuntu 8.04 1T5, Hardy Heron)

1.4.8 Google Play Store

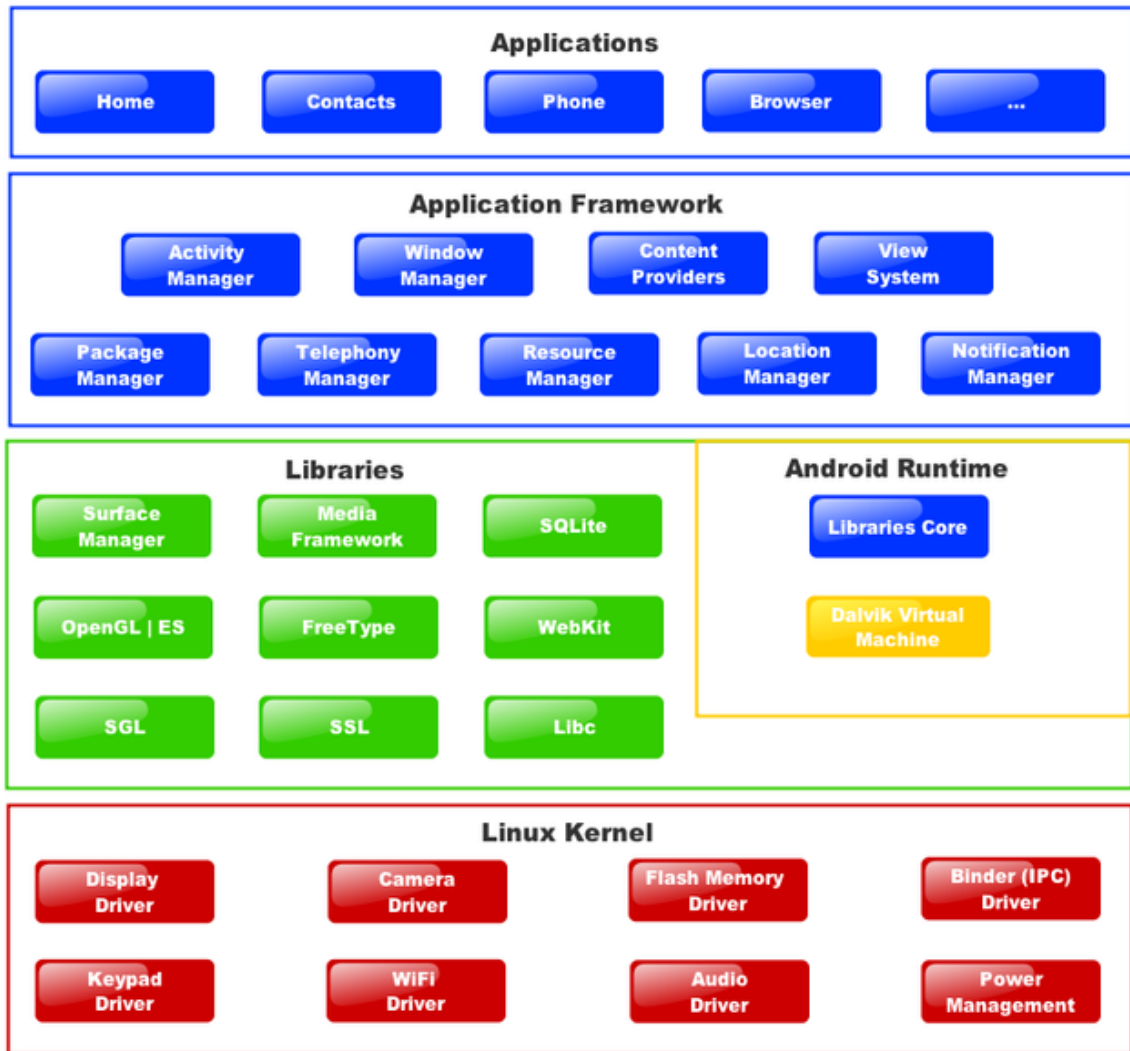
Ένα ακόμη από τα βασικά χαρακτηριστικά του λειτουργικού Android είναι το Google Play, ή αλλιώς πρώην Android Market, είναι μια ψηφιακή πλατφόρμα διανομής εφαρμογών για το λειτουργικό σύστημα Android, η οποία είναι σε απευθείας σύνδεση με τα ηλεκτρονικά και ψηφιακά μέσα αποθήκευσης και λειτουργεί από την Google. Η υπηρεσία επιτρέπει στους χρήστες να περιηγηθούν και να κατεβάσουν εφαρμογές που έχουν αναπτυχθεί με το Android SDK και

δημοσιεύονται μέσω της Google, καθώς και μουσική, περιοδικά, βιβλία, ταινίες και τηλεοπτικά προγράμματα. Οι χρήστες μπορούν επίσης να αγοράσουν υλικό, όπως Chromebooks, το Google Nexus branded κινητών συσκευών, Chromecast, και αξεσουάρ.

Πολλές εφαρμογές μπορούν να απευθύνονται σε συγκεκριμένους χρήστες με βάση ένα συγκεκριμένο χαρακτηριστικό του υλικού της συσκευής τους, όπως ο αισθητήρας κίνησης ή μια κάμερα στο μπροστινό μέρος (για online βίντεο κλήσης). Στις 6 Μαρτίου 2012, με τη συγχώνευση της μουσικής του Android Market και τη Google, η υπηρεσία μετονομάστηκε σε Google Play για να ταυτιστεί με το rebranding της στρατηγικής ψηφιακής διανομής της Google. Οι υπηρεσίες που λειτουργούν κάτω από το Google Play banner είναι: Μουσική Google Play, Google Play Books, Google Play Περίπτερο, Google Play Κινηματογράφος & Τηλεόραση και το Google Play Games. Από τον Ιούλιο του 2013 το κατάστημα Play Google έφτασε επίσημα πάνω από 1 εκατομμύριο δημοσιευμένες εφαρμογές και πάνω από 50 δισεκατομμύρια downloads.

1.5 Αρχιτεκτονική

Για να καταλάβει κάποιος με ποιον ακριβώς τρόπο δουλεύει το Android θα ήταν καλό να κοιτάξει και να μελετήσει την Εικόνα 1.1 στην οποία φαίνονται τα διαφορετικά στρώματα από τα οποία αποτελείται το λειτουργικό σύστημα του Android. Η εικόνα προβάλετε παρακάτω και έπειτα αναλύονται οι βασικοί τομείς στους οποίους χωρίζεται το λειτουργικό σύστημα.



Εικόνα 1.1 : Η αρχιτεκτονική του λειτουργικού Android.

Όπως παρατηρείτε, το λειτουργικό σύστημα του Android χωρίζεται σε 5 τομείς, σε 4 διαφορετικά στρώματα. Αναφέρονται συνοπτικά οι 5 βασικοί τομείς και η λειτουργία που επιτελεί ο καθένας.

🔧 Linux Kernel (Πυρήνας Linux)

Είναι ο πυρήνας στον οποίο βασίζεται το λειτουργικό σύστημα Android. Στο στρώμα αυτό περιέχονται όλοι οι χαμηλού επιπέδου οδηγοί της συσκευής (low level device drivers) που χρειάζονται για τα διάφορα υλικά μέρη της συσκευής.

✚ Βιβλιοθήκες (Libraries)

Οι βιβλιοθήκες περιέχουν όλο τον κώδικα ο οποίος παρέχει τα βασικά χαρακτηριστικά ενός λειτουργικού συστήματος Android.

✚ Android Runtime

Στο ίδιο στρώμα με τις βιβλιοθήκες, ο τομέας Android run time παρέχει ένα σύνολο από βασικές βιβλιοθήκες που δίνουν τη δυνατότητα στους προγραμματιστές να γράψουν εφαρμογές για το Android σε γλώσσα προγραμματισμού Java. Ακόμα, περιέχει την εικονική μηχανή (virtual machine) Dalvik η οποία επιτρέπει σε κάθε εφαρμογή να τρέξει μέσα σε δική της διαδικασία, με δική της ξεχωριστή εικονική μηχανή. Η Dalvik είναι μια ειδική εικονική μηχανή, ειδικά σχεδιασμένη για Android και βελτιστοποιημένη για χρήση με κινητές συσκευές που λειτουργούν με μπαταρία και διαθέτουν περιορισμένες δυνατότητες σε μνήμη και CPU.

✚ Εφαρμογές (Applications)

Σε αυτό το υψηλότερο στρώμα βρίσκονται τόσο οι εφαρμογές οι οποίες διατίθενται μαζί με την συσκευή όπως οι επαφές (Contacts), το Τηλέφωνο (Phone), ο Περιηγητής (Browser κλπ) όσο και οι εφαρμογές που μπορεί να κατεβάσει ο χρήστης από το κεντρικό «κατάστημα εφαρμογών», το Google Play.

1.6 Οι εκδόσεις του Android

Το Android έχει χτιστεί πάνω στο λειτουργικό σύστημα Linux όπως προαναφέρθηκε, με τροποποιήσεις που να ταιριάζουν σε κινητές συσκευές. Από την δημιουργία του Android (2008) μέχρι σήμερα έχουν περάσει πολλές εκδόσεις, η ονομασία των οποίων ακολουθεί παρακάτω (Εικόνα 1.2). Ας δούμε με χρονολογική σειρά τις κυριότερες εκδόσεις Android που κυκλοφόρησαν από το 2008 μέχρι σήμερα με παράλειψη την έκδοση Kit Kat(τελευταία) στην οποία γίνεται εκτενή αναφορά στην συνέχεια :



Εικόνα 1.2 : Διάφορες εκδόσεις Android με το λογότυπο τους.

Αναλυτικά οι εκδόσεις έχουν ως εξής:

✚ Android 1.0 και Android 1.1

Η έκδοση 1.0 πρωτοκυκλοφόρησε το 2008 με το πρώτο Android smart phone HTC Dream.

✚ Android 1.5 “Cupcake”

Κυκλοφόρησε την άνοιξη του 2009 και ήταν το πρώτο που είχε όνομα γλυκού.

✚ Android 1.6 “Donut”

Κυκλοφόρησε το φθινόπωρο του 2009 και συμπεριελάμβανε νέες βελτιώσεις όπως αυτόματη περιστροφή οθόνης και εφέ κίνησης κατά την περιήγηση στις διαφορετικές οθόνες.

✚ Android 2.0“Eclair”

Κυκλοφόρησε μόλις ένα μήνα μετά το Donut (φθινόπωρο 2009) και συμπεριελάμβανε αρκετές μικρές βελτιώσεις.

Android 2.2-2.2.3 “Froyo”

Κυκλοφόρησε το Μάιο του 2010 και είναι συντόμευση της φράσης “Frozen Yogurt” (παγωμένο γιαούρτι). Είναι η πρώτη έκδοση του Android που υποστήριζε Adobe Flash.

Android 2.3-2.3.7 “Gingerbread”

Κυκλοφόρησε τον Δεκέμβρη του 2010 και παραμένει μέχρι και σήμερα η πιο πολυπληθής έκδοση του Android. Ήταν πολύ πιο γρήγορο και εύχρηστο από τις προηγούμενες εκδόσεις και έδινε στους δημιουργούς εφαρμογών μεγαλύτερες δυνατότητες.

Android 3.0-3.2 “Honeycomb”

Κυκλοφόρησε τον Φεβρουάριο του 2011 και ήταν διαθέσιμη μόνο για tablets. Η πρώτη συσκευή που έτρεχε σε Honeycomb ήταν το tablet Motorola Xoom.

Android 4.0-4.0.2 “Ice Cream Sandwich”

Κυκλοφόρησε τον Οκτώβρη του 2011 και έφερε πάρα πολλές αλλαγές στο λειτουργικό σύστημα. Μερικές από αυτές είναι η δυνατότητα χρήσης “μαλακών” κουμπιών δηλαδή κουμπιών πάνω στην οθόνη (πίσω, αρχική, κλπ) ενώ μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά. Άλλες δυνατότητες ήταν το Face Unlock, καλύτερη χρήση των φωνητικών εντολών, αναδιαμόρφωση του περιβάλλοντος χρήσης, βελτίωση της ταχύτητας απόκρισης κλπ

Android 4.1-4.2 “Jelly Bean”

Κυκλοφόρησε τον Ιούνιο του 2012 και αποτελεί την καλύτερη έκδοση του Android μέχρι σήμερα. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορα και καλοφτιαγμένα από ποτέ ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην χρήση φωνής (υπαγόρευση κειμένου) και στην κάμερα.

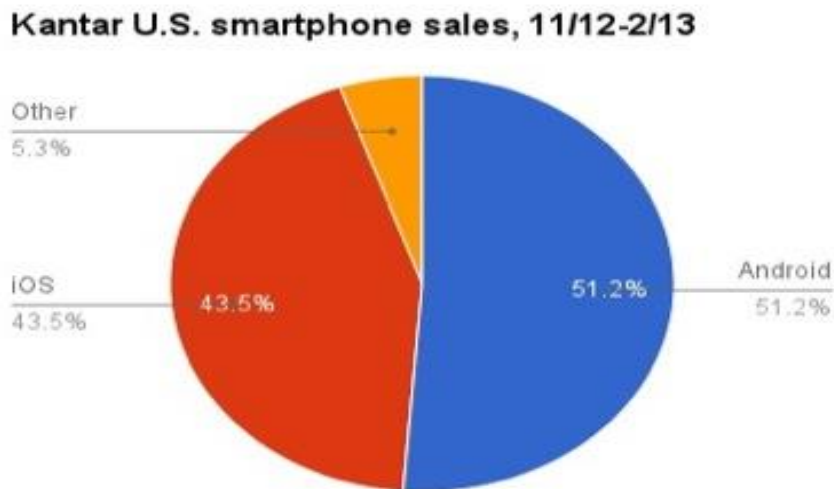
Android 4.4 “Kit Kat”

Το Android 4.4 ονομάζεται Kit Kat και φυσικά θα θυμίζει-διαφημίζει το ομώνυμο σοκολατένιο προϊόν. Αυτό είναι ουσιαστικά το περιεχόμενο της εμπορικής συμφωνίας της Google με την εταιρεία Nestlé, που παράγει τις γνωστές σοκολατένιες μπάρες. Μάλιστα, οι δύο εταιρείες κυκλοφόρησαν ήδη τα διαφημιστικά σποτ για το πάντρεμα του mobile λειτουργικού της Google με τις σοκολάτες Kit Kat.

Το Android 4.4 Kit Kat είναι εμφανώς πιο καθαρό και απλούστερο από την προηγούμενη έκδοση, παρόλα αυτά όμως οι αλλαγές δεν είναι συγκλονιστικές. Μια ενδιαφέρουσα αλλαγή θα παρατηρήσετε στη lock screen όταν ακούτε μουσική, καθώς θα σας εμφανίζει το εξώφυλλο του album και θα έχετε τη δυνατότητα να αλλάξετε τραγούδι χωρίς να ξεκλειδώσετε τη συσκευή. Εξάλλου, όταν ο χρήστης διαβάζει κάποιο e-book ή παρακολουθεί video θα κρύβονται αυτομάτως όλα τα περιττά στοιχεία για καλύτερη θέαση.

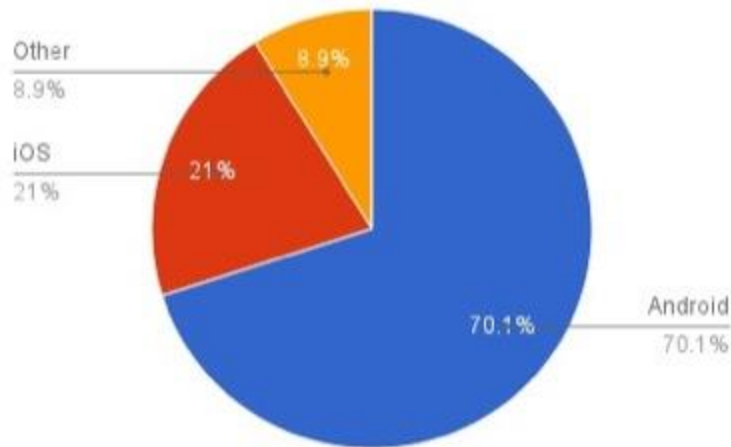
1.7 Android vs. iOS

Η ατέρμονη διαμάχη μεταξύ iOS και Android είναι από αυτές τις περιπτώσεις που κάθε πλευρά θα καταθέσει τα δικά της επιχειρήματα αλλά στην πραγματικότητα το μόνο που μετράει είναι η οπτική γωνία που προσεγγίζει κανείς το θέμα. Μπορεί το Android για παράδειγμα να έχει κατακλύσει την αγορά και να κυριαρχεί σε ποσότητα αλλά όταν η Apple ταυτόχρονα έχει την πλειοψηφία των κερδών από τις συσκευές και το οικοσύστημα που πουλάει. Ο Harry McCracken (Αρθογράφος στο TIME.com) αποφάσισε να συγκεντρώσει όλα τα στατιστικά για τις δύο πλατφόρμες του οποίου η έρευνα φανερώνεται παρακάτω:



Εικόνα 1.3 : Στατιστικά στοιχεία πωλήσεων smart phone στις ΗΠΑ

IDC worldwide smartphone shipments, Q4 2012



Εικόνα 1.3 : Στατιστικά στοιχεία πωλήσεων smart phone παγκοσμίως.

1.8 Επίλογος

Συνοψίζοντας λοιπόν, σε αυτό το κεφάλαιο έγινε μία εισαγωγή στο λειτουργικό σύστημα Android προκειμένου ο αναγνώστης αυτής της πτυχιακής εργασίας να κατανοήσει τι είναι τελικά το Android με πολλαπλές αναφορές σε ορισμούς, στην ιστορική του αναδρομή, στα χαρακτηριστικά του αλλά και στην εξέλιξη του μέχρι και σήμερα. Επιπρόσθετα σε αυτό το κεφάλαιο δεν θα μπορούσε να παραλειφτεί η αρχιτεκτονική του Android και τα μέρη του πυρήνα που απαρτίζεται καθώς και στον ορισμό εξειδικευμένων όρων και τεχνολογιών πάνω στην ανάπτυξη τέτοιου είδους εφαρμογών.

2 Προγραμματισμός και ανάπτυξη εφαρμογών σε περιβάλλον Android

2.1 Εισαγωγή

Ίσως αυτό το κεφάλαιο είναι ένα από τα πιο σημαντικά κεφάλαια για κάποιον που τώρα αρχίζει τον προγραμματισμό και την ανάπτυξη εφαρμογών στο περιβάλλον του Android και αυτό γιατί θα γίνει μία αναλυτική περιγραφή για το τι κατηγορίες εφαρμογών υπάρχουν στο Android καθώς και ανάλυση των συστατικών στοιχείων μιας εφαρμογής που είναι απαραίτητα να κατανοηθούν πριν μεταβούμε στην δημιουργία της εφαρμογής (προσφορά και ζήτηση αγοράς εργασίας). Θα γίνει πλήρη αναφορά στο τι εννοούμε με την έννοια Activity, Fragment κ.α που είναι ο θεμελιώδης λίθος στον προγραμματισμό Android εφαρμογών.

2.2 Κατηγορίες Εφαρμογών

Σε αυτό το σημείο θα αναφερθούν και θα αναλυθούν τα 4 είδη εφαρμογών που μπορούν να αναπτυχθούν στο λειτουργικό σύστημα του Android.

2.2.1 Εφαρμογές προσκηνίου

Μια εφαρμογή που χρησιμοποιείται μόνο όταν είναι στο προσκήνιο και αναστέλλεται όταν δεν είναι ορατή ανήκει σε αυτήν την κατηγορία. Όταν αναπτύσσουμε εφαρμογές προσκηνίου (Foreground applications) πρέπει να έχουμε κατά νου τον κύκλο ζωής ενός Activity έτσι ώστε να μπορεί να εναλλάσσεται μεταξύ του παρασκηνίου και του προσκηνίου. Οι εφαρμογές αυτού του είδους δεν έχουν τον έλεγχο του κύκλου ζωής τους κάτι που σημαίνει ότι πρέπει να αποθηκεύουμε την κατάσταση της εφαρμογής όταν αυτή απομακρυνθεί από το προσκήνιο και στη συνέχεια όταν επανέλθει σε αυτό να την επαναφέρουμε στην ίδια κατάσταση. Επίσης είναι πολύ σημαντικό, οι εφαρμογές που ανήκουν σε αυτή την κατηγορία να παρουσιάζουν μια αρκετά καλή διεπαφή χρήστη.

2.2.2 Εφαρμογές παρασκηνίου

Οι εφαρμογές που ανήκουν σε αυτήν την κατηγορία έχουν ελάχιστη ή και καθόλου αλληλεπίδραση με το χρήστη και πέραν της ρύθμισής τους τον περισσότερο χρόνο εκτέλεσής τους παραμένουν κρυφές. Τις περισσότερες φορές παρακολουθούν για μηνύματα ή ενέργειες που προκαλούνται από το υλικό, το σύστημα ή άλλες εφαρμογές αντί να βασίζονται στην αλληλεπίδραση με το χρήστη. Μπορούμε να δημιουργήσουμε εντελώς άορατες υπηρεσίες, αλλά στην πράξη είναι καλό να παρέχουμε ένα βασικό επίπεδο ελέγχου στο χρήστη. Κατ'ελάχιστο λοιπόν είναι καλό να επιτρέπεται στους χρήστες να επιβεβαιώσουν ότι η υπηρεσία τρέχει και να τους αφήνουμε να ρυθμίζουν, να παύουν ή να την τερματίζουν όποτε επιθυμούν.

2.2.3 Διακοπτόμενες εφαρμογές

Υπάρχουν περιπτώσεις που χρειάζεται μια εφαρμογή να δέχεται εισόδους από το χρήστη αλλά και να αντιδρά σε γεγονότα όταν δεν είναι στο προσκήνιο. Οι εφαρμογές της κατηγορίας αυτής λοιπόν είναι μια ένωση ορατών Activities και άορατων υπηρεσιών παρασκηνίου ή broadcast receivers.

Τέτοιες εφαρμογές πρέπει να γνωρίζουν ανά πάσα στιγμή την κατάστασή τους όταν αλληλεπιδρούν με το χρήστη. Για παράδειγμα θα πρέπει να ενημερώνουν το UI όταν είναι ορατό και να στέλνουν notifications για να ενημερώνουν το χρήστη όταν είναι στο παρασκήνιο. Παραδείγματα τέτοιων εφαρμογών είναι αυτές για email ή για τη λήψη ειδήσεων.

2.2.4 Widgets & Live Wallpapers

Δημιουργώντας μια τέτοια εφαρμογή Android, παρέχονται διαδραστικά εικονικά στοιχεία με τα οποία είναι δυνατή η πρόσθεση λειτουργικότητας στα home screens. Οι εφαρμογές μόνο με widgets είναι χρήσιμες στην παροχή δυναμικής πληροφορίας όπως για παράδειγμα τη στάθμη της μπαταρίας ή την πρόβλεψη του καιρού.

2.3 Συστατικά στοιχεία μιας εφαρμογής

Σε αυτό το μέρος της πτυχιακής εργασίας θα ήταν σκόπιμο να αναλυθούν τα συστατικά στοιχεία και η διατύπωση της ορολογίας που χρησιμοποιείται στην ανάπτυξη εφαρμογών και που παρέχει μια πιο διεξοδική ανάλυση του τρόπου, που οι εφαρμογές Android λειτουργούν και αλληλεπιδρούν μεταξύ τους.

2.3.1 Context (Περιβάλλον)

Το context είναι το κέντρο εντολών για μία εφαρμογή Android. Όλες οι λειτουργίες που αφορούν στην εφαρμογή μπορούν να προσπελαστούν μέσω του context.

Το Context της εφαρμογής είναι η κεντρική θέση για όλες τις λειτουργίες εφαρμογών ανωτέρου επιπέδου. Η κλάση Context μπορεί να χρησιμοποιηθεί για την διαχείριση λεπτομερειών ρύθμισης παραμέτρων για συγκεκριμένες εφαρμογές, όπως για λειτουργίες και δεδομένα για μια ολόκληρη εφαρμογή. Χρησιμοποιούμε το Context της εφαρμογής για να προσπελάσουμε ρυθμίσεις και πόρους που μοιράζονται σε πολλαπλά στιγμιότυπα Activity. Η ανάκτηση του Context της εφαρμογής για την τρέχουσα διεργασία επιτυγχάνεται χρησιμοποιώντας την μέθοδο `getApplicationContext()` ως εξής:

`Context context= getApplicationContext();`

Αφού γίνει η ανάκτηση ενός έγκυρου Context της εφαρμογής, μπορεί να χρησιμοποιηθεί για προσπέλαση χαρακτηριστικών και υπηρεσιών για ολόκληρη την εφαρμογή.

2.3.2 Activity (Δραστηριότητα)

Το λειτουργικό σύστημα Android χρησιμοποιεί ένα ασύγχρονο μηχανισμό αποστολής/παραλαβής μηνυμάτων ώστε να ανταπεξέλθει σε αιτήσεις εργασιών με την κατάλληλη Δραστηριότητα (Activity). Πρακτικά κάθε τέτοια αίτηση είναι ένα μήνυμα που δηλώνει μία πρόθεση για να γίνει κάτι. Για παράδειγμα, όταν αναφερόμαστε στο πάτημα ενός κουμπιού ουσιαστικά αυτό προγραμματιστικά θα γίνει με την χρήση Intent .

Οι activities αποτελούν το βασικότερο στοιχείο μιας εφαρμογής. Κάθε activity ουσιαστικά αντιπροσωπεύει μια οθόνη μαζί με την διεπιφάνεια χρήστη. Μια εφαρμογή συνήθως αποτελείται από πολλές activities οι οποίες είναι χαλαρά δεμένες μεταξύ τους. Συνηθέστερα, μια activity αποτελεί την main (βασική) activity η οποία είναι και η οθόνη που βλέπει ο χρήστης όταν ανοίγει την εφαρμογή για πρώτη φορά. Κάθε activity μπορεί να καλέσει μια νέα activity για να εκτελέσει διαφορετικές λειτουργίες. Κάθε φορά που μια νέα activity ξεκινάει, η προηγούμενη activity σταματάει. Το σύστημα όμως την διατηρεί και την αποθηκεύει σε μια στοίβα (LIFO), το λεγόμενο “**backstack**”. Όταν λοιπόν μια νέα activity ξεκινάει, τότε εμφανίζεται στην οθόνη και αποθηκεύεται τελευταία στο **back stack**. Έτσι, όταν ο χρήστης τελειώσει με τη χρήση της και πατήσει το πλήκτρο Back, τότε αυτή γίνεται pop από την στοίβα, καταστρέφεται και η οθόνη επανέρχεται στην προηγούμενη activity.

Αν πατηθεί το πλήκτρο Back ενώ ο χρήστης βρίσκεται στην main activity τότε εξέρχεται από την εφαρμογή. Για να κατασκευαστεί μια activity, πρέπει να δημιουργηθεί μια υποκλάση της βασικής κλάσης Activity. Σε αυτή την υποκλάση πρέπει να εφαρμοστούν συγκεκριμένες μέθοδοι τις οποίες καλεί το σύστημα όταν η activity μεταβαίνει σε διαφορετικά στάδια του κύκλου ζωής της. Τέτοια στάδια είναι όταν η activity δημιουργείται (create), σταματάει (stop), επανέρχεται (resume) ή καταστρέφεται (destroy). Οι δύο πιο βασικές τέτοιες μέθοδοι είναι:

 onCreate()

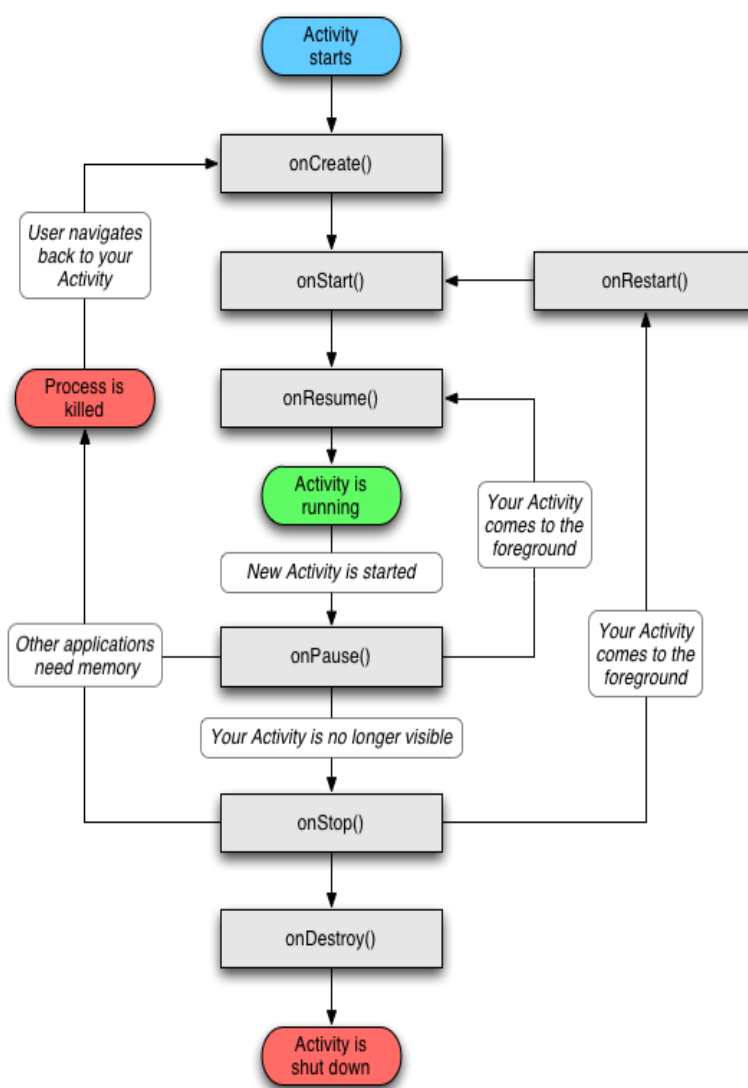
Πρέπει αναγκαστικά να εφαρμοστεί αυτή η μέθοδος. Το σύστημα την καλεί όταν δημιουργεί την activity. Εδώ πρέπει να αρχικοποιηθούν τα απαραίτητα στοιχεία της activity. Το πιο σημαντικό είναι να γίνει κλήση σε αυτό το σημείο στην μέθοδο **setContentView()** με την οποία ορίζεται η διάταξη (layout) της διεπιφάνειας χρήστη (user interface) αυτής της activity.

 onPause()

Το σύστημα καλεί αυτή τη μέθοδο όταν έχει ενδείξεις ότι ο χρήστης φεύγει από την activity (χωρίς να σημαίνει η activity καταστρέφεται). Εδώ συνήθως ο

προγραμματιστής πρέπει να κάνει όποιες αλλαγές θα χρειαστεί να παραμείνουν και μετά το πέρας αυτής της περιόδου γιατί ο χρήστης μπορεί να μην επιστρέψει.

Υπάρχουν και άλλες βασικές μέθοδοι πέρα από αυτές, οι οποίες χρησιμοποιούνται από το σύστημα και που ο προγραμματιστής θα πρέπει να χρησιμοποιεί ώστε να προσφέρει στον χρήστη μια ομαλή εμπειρία χρήσης. Ο βασικός κύκλος ζωής μιας activity φαίνεται είναι η Εικόνα 2.1 που προβάλετε αμέσως μετά.



Εικόνα 2.1 : Κύκλος ζωής μίας Δραστηριότητας(Activity).

2.3.3 Intent (Πρόθεση)

Το λειτουργικό σύστημα Android χρησιμοποιεί ένα ασύγχρονο μηχανισμό αποστολής/παραλαβής μηνυμάτων ώστε να ανταπεξέλθει σε αιτήσεις εργασιών με την κατάλληλη Δραστηριότητα (Activity). Κάθε αίτηση συσκευάζεται ως μία Intent. Αυτό σημαίνει ότι κάθε τέτοια αίτηση, είναι ένα μήνυμα που δηλώνει μία πρόθεση για να γίνει κάτι. Για παράδειγμα όταν αναφερόμαστε στο πάτημα ενός κουμπιού ουσιαστικά αυτό προγραμματιστικά θα γίνει με την χρήση Intent .

2.3.4 Service (Υπηρεσία):

Εργασίες που δεν απαιτούν την αλληλεπίδραση του χρήστη μπορούν να ενθυλακωθούν σε μία υπηρεσία. Μία υπηρεσία είναι περισσότερο χρήσιμη όταν οι λειτουργίες διαρκούν πολύ (παίρνοντας το βάρος από τη χρονοβόρα επεξεργασία) ή πρέπει να πραγματοποιούνται τακτικά (όπως ο έλεγχος ενός διακομιστή για νέα αλληλογραφία).

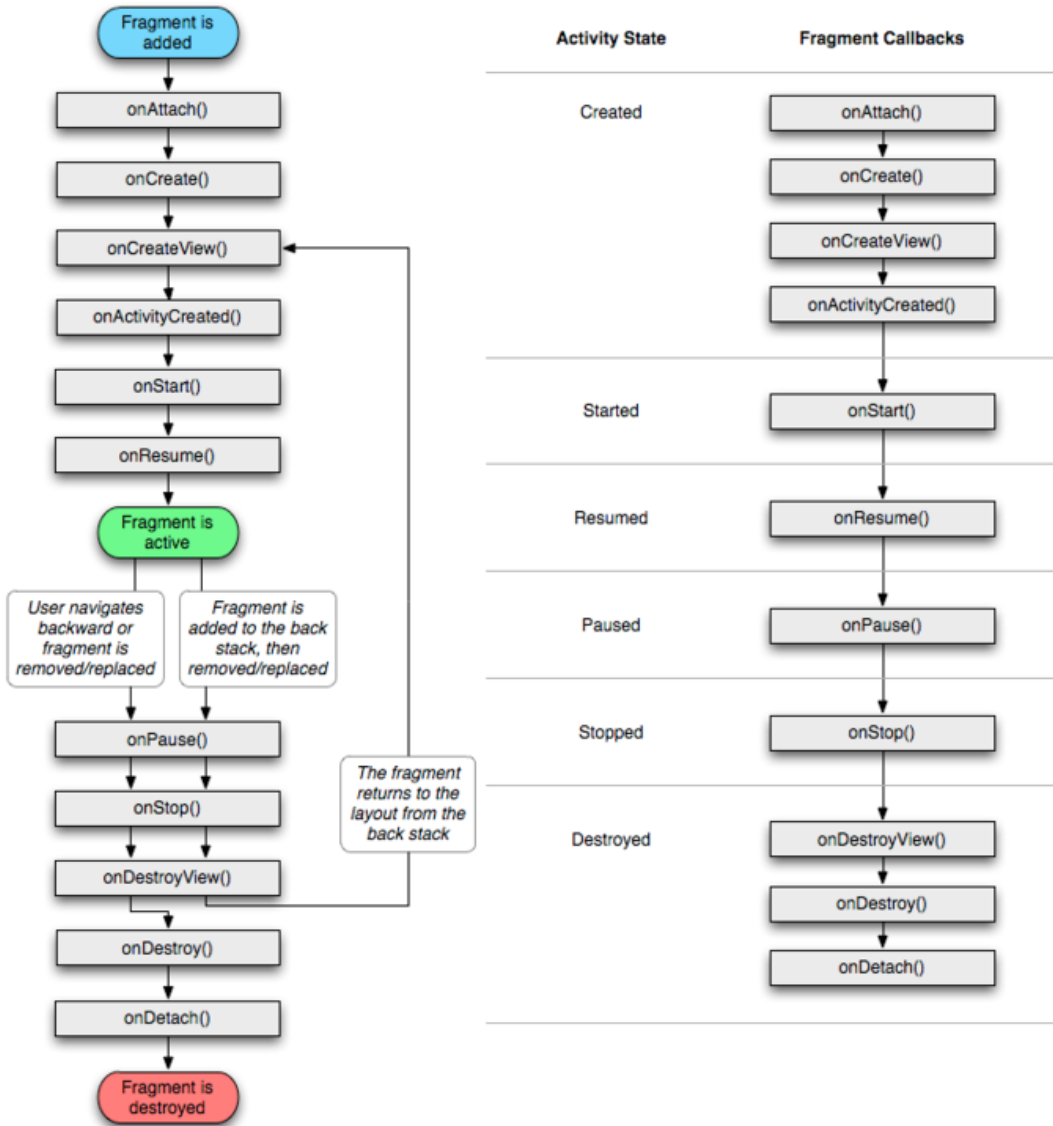
2.3.5 Fragments

Τα Fragments παρά το ότι είναι ένα εξειδικευμένο στοιχείο στην ανάπτυξη εφαρμογών για Android , είναι αναγκαίο να αναφερθούν στην παρούσα πτυχιακή εργασία διότι έχουν χρησιμοποιηθεί κατά κόρον στην ανάπτυξη της συγκεκριμένης εφαρμογής. Επίσης, είναι ένα αρκετά νέο στοιχείο στον κόσμο του Android καθ' ότι έγινε διαθέσιμο από το API 11 και μετά, δηλαδή από την έκδοση 3.0. Από τότε χρησιμοποιούνται ευρύτατα από τους προγραμματιστές μιας και είναι πάρα πολύ χρήσιμα και ευέλικτα. Τα Fragments αναπαριστούν τη συμπεριφορά ή ένα μέρος της διεπιφάνειας χρήστη μέσα σε μια Activity.

Μπορούν να συνδυαστούν πολλά fragments σε μια μόνο activity ώστε να κατασκευαστεί ένα UI με πολλά παράθυρα και επίσης μπορεί να επαναχρησιμοποιηθεί το ίδιο fragment σε πολλές activities.

Ουσιαστικά, ένα fragment είναι κατασκευαστικό κομμάτι μιας activity, με το δικό του κύκλο ζωής, την δική του είσοδο δεδομένων και το οποίο μπορεί να προστεθεί ή να αφαιρεθεί κατά την διάρκεια που μία εφαρμογή εκτελείται.

Ένα fragment πρέπει πάντα να είναι ενσωματωμένο μέσα σε μια activity και ο κύκλος ζωής του επηρεάζεται άμεσα από τον κύκλο ζωής αυτής της activity (Δραστηριότητα). Για παράδειγμα, όταν η activity σταματάει, το ίδιο συμβαίνει και σε όλα τα fragments μέσα σε αυτή. Όταν η activity καταστρέφεται, καταστρέφονται και όλα τα fragments σε αυτήν. Όμως, το σημαντικό κομμάτι είναι ότι, καθώς η activity εκτελείται, ο προγραμματιστής μπορεί να χειριστεί κάθε fragment ξεχωριστά, να προσθέσει ή να αφαιρέσει διαφορετικά fragments, ακόμα και να αντικαταστήσει fragments με άλλα fragments αυτό ονομάζεται fragment transaction. Όπως και με τις activities, έτσι και τα fragments, μπορούν να τα προστεθούν στο Back Stack. Το Back Stack αυτό χρησιμοποιείται από την δραστηριότητα (Activity) και δίνει τη δυνατότητα στο χρήστη να αντιστρέψει μια fragment transaction, να προηγηθεί δηλαδή προς τα πίσω με τη χρήση του πλήκτρου Back. Ο κύκλος ζωής του fragment είναι ανάλογος της παρακάτω εικόνας.



Εικόνα 2.2: Κύκλος ζωής του Fragment .

2.3.6 Broadcast Receivers

Οι Broadcast Receivers είναι ένα στοιχείο το οποίο αποκρίνεται σε ανακοινώσεις που γίνονται σε επίπεδο λειτουργικού συστήματος. Για παράδειγμα το κλείσιμο της οθόνης της συσκευής ή το μικρό ποσοστό της μπαταρίας ή ακόμα και το τράβηγμα μιας φωτογραφίας, αποτελούν Broadcast ανακοινώσεις στο σύστημα. Ένας Broadcast Receiver μπορεί να ενεργεί με την έλευση τέτοιων αντίστοιχων συμβάντων.

2.3.7 Processes and Threads (Διαδικασίες και Νήματα)

Τα σύγχρονα υπολογιστικά συστήματα από επιτραπέζιους υπολογιστές μέχρι υπέρ-υπολογιστές φέρουν πολυπύρηνους (multicore) επεξεργαστές. Οι πολυπύρνηνοι επεξεργαστές αποδίδουν περισσότερη υπολογιστική ισχύ σε ένα τσιπ και το λογισμικό διαδραματίζει καθοριστικό ρόλο στην αξιοποίησή τους. Ο πολυνηματικός προγραμματισμός είναι ένα ευρέως χρησιμοποιούμενο πρότυπο προγραμματισμού για την αξιοποίηση του ταυτοχρονισμού σε πολυπύρηννα συστήματα.

Στην υλοποίηση της εφαρμογής που θα αναλυθεί αναλυτικά στα επόμενα κεφάλαια, χρησιμοποιείται ο πολυνηματικός προγραμματισμός κατά κόρον και θα ήταν εύστοχο σε αυτό το σημείο να αναφερθούν τα είδη του, προσαρμοσμένα πάνω στην πλατφόρμα του Android. Τα είδη αυτά είναι:

2.3.7.1 GUI Thread

Το UIThread είναι το κύριο νήμα της εκτέλεσης μιας εφαρμογής. Όλα τα συστατικά μιας εφαρμογής (Δραστηριότητες, Υπηρεσίες, ContentProviders, Broadcast Receivers) δημιουργούνται σε αυτό το νήμα, καθώς και οποιοδήποτε σύστημα κλήσεων προς τα συστατικά αυτά, εκτελούνται σε αυτό το νήμα. Όταν αναπαράγεται ένα νέο νήμα για να κάνει τη δουλειά στο παρασκήνιο, αυτός ο κώδικας δεν εκτελείται στο UIThread. Έτσι τι συμβαίνει εάν αυτό το νήμα πρέπει να κάνει κάτι που αλλάζει το UI; Αυτό που πρέπει να γίνει είναι να κληθεί η `runOnUiThread` μέθοδος.

2.3.7.2 AsyncTask

Οι χρήστες απαιτούν εφαρμογές με καλή ανταπόκριση, που σημαίνει ότι οι χρονοβόρες ενέργειες όπως η δικτύωση δεν θα πρέπει να μπλοκάρουν και κατ' επέκταση να καθυστερούν το κύριο νήμα της διεπαφής χρήστη (UI). Το συλ δικτύωσης το οποίο έχει παρουσιαστεί ως τώρα αναγκάζει το νήμα UI στο οποίο εκτελείται να σταματήσει μέχρι να ολοκληρωθεί η ενέργεια. Για μικρές εργασίες, αυτό ίσως να είναι αποδεκτό. Ωστόσο, όταν ο χρόνος λήγει, μεγάλες ποσότητες δεδομένων ή μεγάλη ποσότητα πρόσθετης επεξεργασίας, όπως η συντακτική ανάλυση XML, προστίθενται στο σύνολο και γι' αυτό θα πρέπει να μεταφέρθούν αυτές οι χρονοβόρες ενέργειες έξω απ' το κύριο νήμα UI.

Η κλάση AsyncTask είναι μία ειδική κλάση για ανάπτυξη εφαρμογών με το Android η οποία παρέχει επεξεργασία στο παρασκήνιο και συμβάλλει στην διευκόλυνση της επικοινωνίας με το νήμα UI ενώ γίνεται διαχείριση του κύκλου ζωής της εργασίας του παρασκηνίου μέσα στο πλαίσιο του κύκλου ζωής της δραστηριότητας. Οι προγραμματιστές μπορούν επίσης να κατασκευάσουν τις δικές τους λύσεις αλληλουχίας ενεργειών χρησιμοποιώντας τυπικές μεθόδους και κλάσεις της Java.

Δύο από τις πιο σημαντικές μεθόδους της κλάσης AsyncTask που αξίζει να αναφερθούν είναι η μέθοδος ***doInBackground()*** η οποία χειρίζεται την επεξεργασία παρασκηνίου ενώ η ***publishProgress()*** ενημερώνει το νήμα UI κατά διαστήματα σχετικά με την πρόοδο της επεξεργασίας παρασκηνίου. Όταν η επεξεργασία παρασκηνίου ολοκληρωθεί, η μέθοδος ***onPostExecute()*** εκτελείται στο νήμα UI, ώστε να προσφέρει μία τελική ενημέρωση.

2.3.8 User Interface

Οι περισσότερες εφαρμογές αναπόφευκτα χρειάζονται κάποιου είδους διεπαφής χρήστη. Σε αυτό το σημείο λοιπόν θα αναλυθούν τα στοιχεία διεπαφών χρήστη που διατίθενται στο πακέτο ανάπτυξης λογισμικού του Android. Κάποια από αυτά τα στοιχεία εμφανίζουν πληροφορίες στον χρήστη, ενώ άλλα συλλέγουν πληροφορίες από αυτόν.

2.3.8.1 Layouts

Τα Layouts είναι .xml αρχεία στα οποία ο προγραμματιστής εφαρμογών γράφει κώδικα όσο αφορά την αλληλεπίδραση του χρήστη με το UI της εφαρμογής. Μερικά από αυτά τα κύρια σχεδιαστικά πρότυπα αναφέρονται παρακάτω.

2.3.8.1.1 *FrameLayout*

Μια εφαρμογή *FrameLayout* σχεδιάστηκε, ώστε να εμφανίζει μία στοιβή θυγατρικών στοιχείων *View*. Υπάρχει η δυνατότητα πρόσθεσης πολλών προβολών σε αυτήν την διάταξη αλλά κάθε *view* σχεδιάστηκε από την πάνω αριστερή γωνία της διάταξης. Μπορεί να χρησιμοποιηθεί ώστε να εμφανίζονται πολλαπλές εικόνες μέσα στην ίδια περιοχή. Τα χαρακτηριστικά διάταξης που διατίθενται για θυγατρικά αντικείμενα *view* του *FrameLayout* υπάρχουν διαθέσιμα στο *Android.control.FrameLayout.LayoutParams*. Η εικόνα 2.3 περιγράφει μερικά από τα σημαντικότερα χαρακτηριστικά που αφορούν συγκεκριμένα σε προβολές *FrameLayout*.

neLayout.LayoutParams

layout.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
<code>top</code>	0x30	Push object to the top of its container, not changing its size.
<code>bottom</code>	0x50	Push object to the bottom of its container, not changing its size.
<code>left</code>	0x03	Push object to the left of its container, not changing its size.
<code>right</code>	0x05	Push object to the right of its container, not changing its size.
<code>center_vertical</code>	0x10	Place object in the vertical center of its container, not changing its size.
<code>fill_vertical</code>	0x70	Grow the vertical size of the object if needed so it completely fills its container.
<code>center_horizontal</code>	0x01	Place object in the horizontal center of its container, not changing its size.
<code>fill_horizontal</code>	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
<code>center</code>	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
<code>fill</code>	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
<code>clip_vertical</code>	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
<code>clip_horizontal</code>	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
<code>start</code>	0x00800003	Push object to the beginning of its container, not changing its size.
<code>end</code>	0x00800005	Push object to the end of its container, not changing its size.

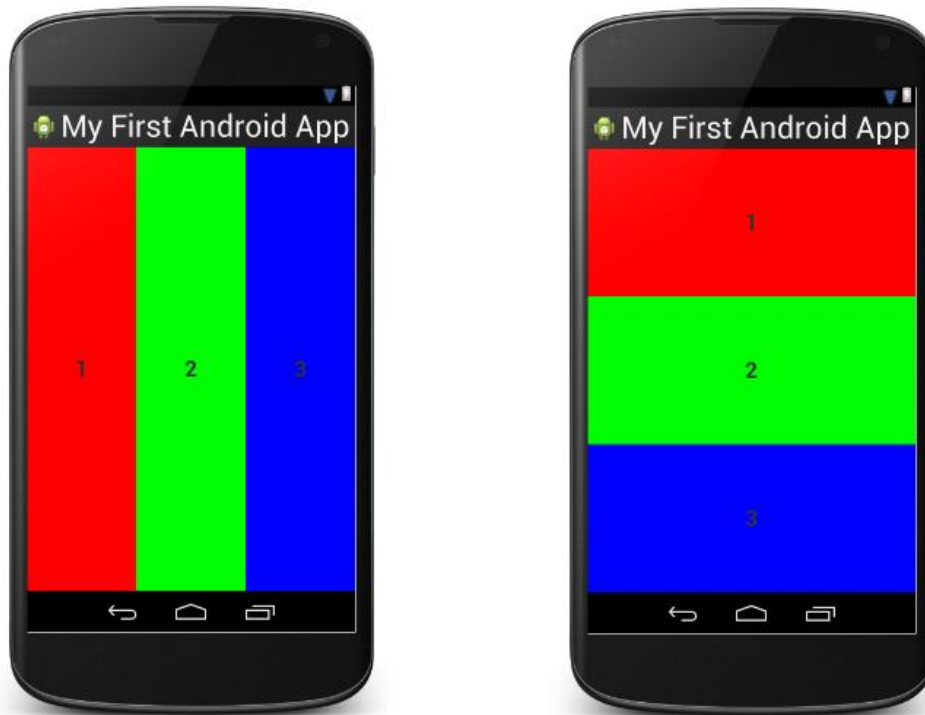
This corresponds to the global attribute resource symbol `layout_gravity`.

Related Methods

Εικόνα 2.3: Χαρακτηριστικά του *FrameLayout* .

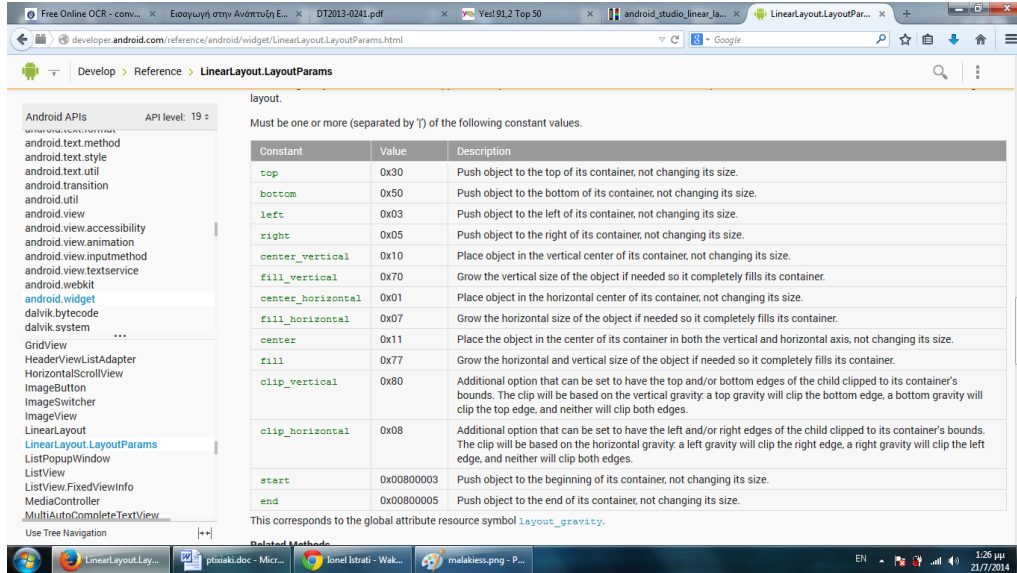
2.3.8.1.2 *LinearLayout*

Μια προβολή *LinearLayout* οργανώνει τα θυγατρικά αντικείμενα της *view* σε μια γραμμή ή σε μία στήλη, ανάλογα από το αν το χαρακτηριστικό προσανατολισμού (*orientation*) στο *.xml* αρχείο είναι κατακόρυφο ή οριζόντιο αντίστοιχα. Αυτή η μέθοδος διάταξης είναι πολύ χρήσιμη για τη δημιουργία φορμών. Μία χαρακτηριστική εικόνα της *Linear* διάταξης είναι η παρακάτω.



Εικόνα 2.4: Παράδειγμα *Linear* διάταξης στο *Android*

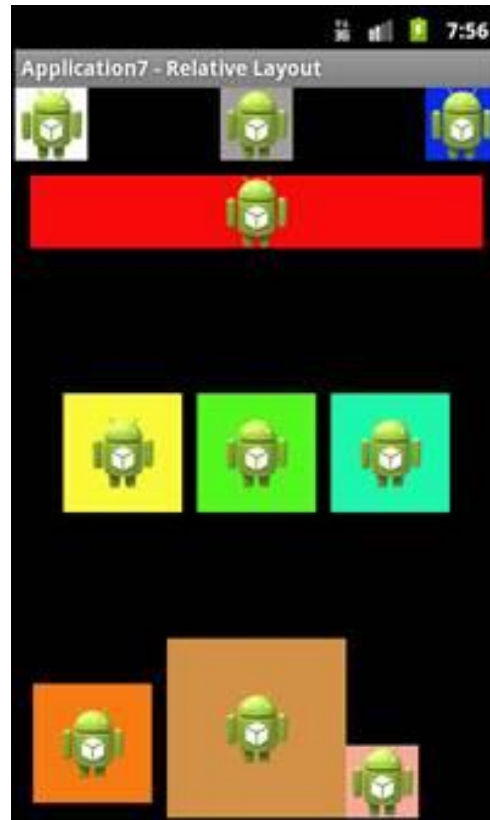
Η εικόνα 2.5 περιγράφει μερικά από τα σημαντικότερα χαρακτηριστικά που αφορούν συγκεκριμένα σε προβολές *LinearLayout* όπως μας τα παραθέτει το *official Documentation* του *Android*.



Εικόνα 2.5: Χαρακτηριστικά Linear Layout στο Android.

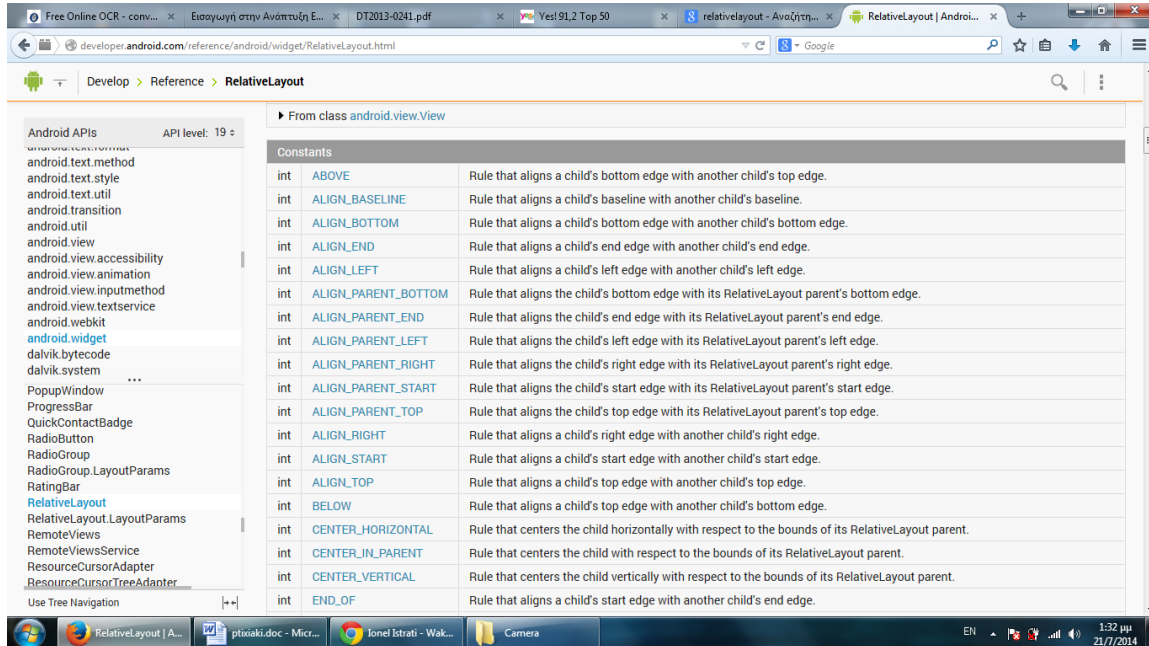
2.3.8.1.3 RelativeLayout

Η προβολή RelativeLayout επιτρέπει να καθορίζεται η θέση των στοιχείων ελέγχου child view μεταξύ τους. Για παράδειγμα, μπορεί να οριστεί ότι ένα θυγατρικό view θα τοποθετηθεί «πάνω» ή «κάτω» ή «στα αριστερά» ή «στα δεξιά» ενός άλλου view, απλά μόνο με το μοναδικό αναγνωριστικό του (ID). Μπορούν επίσης να στοιχιστούν θυγατρικά αντικείμενα view το ένα σε σχέση με το άλλο ή τις πλευρές της γονικής διάταξης. Ο συνδυασμός χαρακτηριστικών RelativeLayout μπορεί να απλοποιήσει τη δημιουργία ενδιαφερόντων διεπαφών χρήστη χωρίς να είναι ανάγκη η καταφυγή σε πολλαπλές ομάδες διάταξης. Η Εικόνα 2.6 δείχνει πώς σχετίζονται μεταξύ τους τα στοιχεία ελέγχου κουμπιών.



Εικόνα 2.6: Παράδειγμα Relative διάταξης στο Android

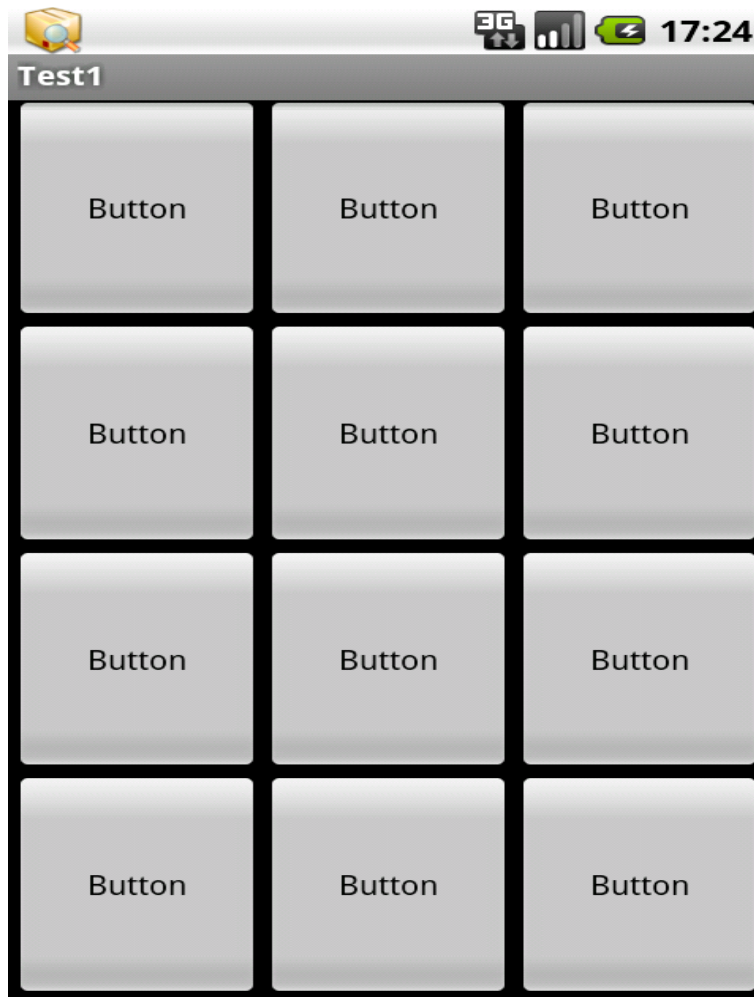
Η εικόνα 2.7 παρακάτω περιγράφει μερικά από τα σημαντικότερα χαρακτηριστικά που αφορούν συγκεκριμένα σε προβολές RelativeLayout όπως μας τα παραθέτει το official Documentation του Android.



Εικόνα 2.7: Χαρακτηριστικά του RelativeLayout .

2.3.8.1.4 TableLayout

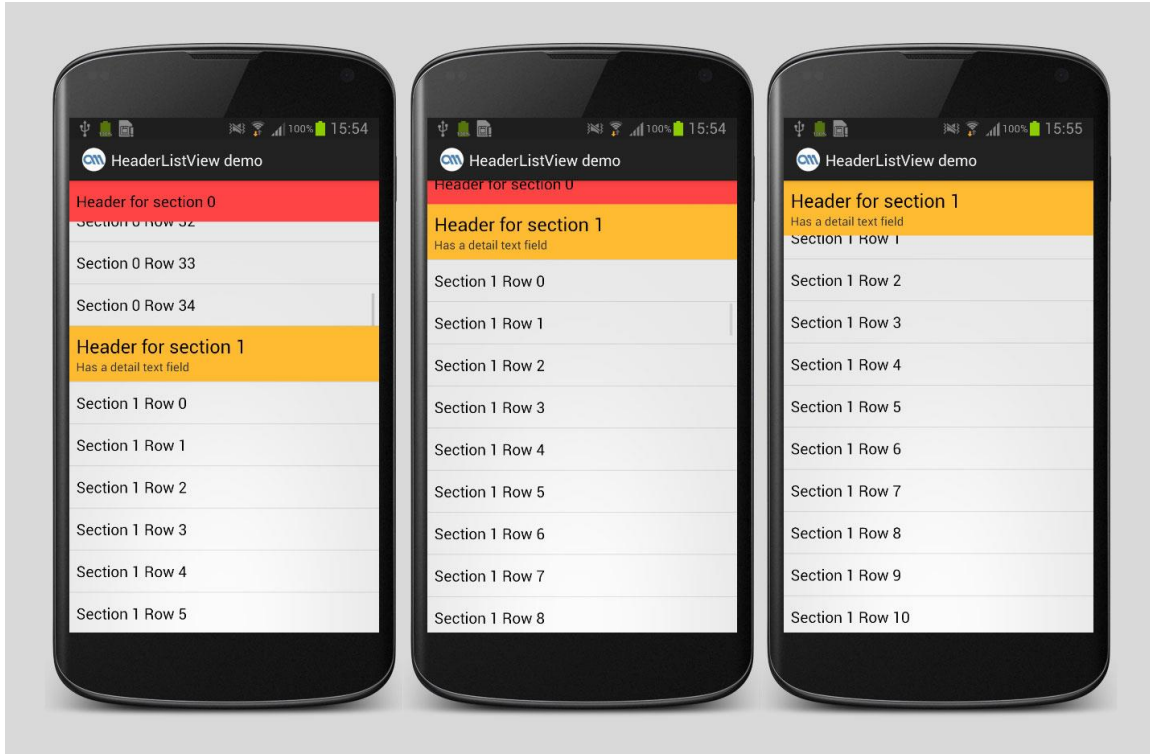
Μία προβολή TableLayout οργανώνει τα παιδιά σε γραμμές, όπως παρουσιάζεται στην Εικόνα 2.8. Εφικτή είναι ακόμη η πρόσθεση μεμονωμένων αντικειμένων view μέσα σε κάθε γραμμή του πίνακα χρησιμοποιώντας ένα View διάταξης Table Row (το οποίο είναι ουσιαστικά ένα οριζόντιο LinearLayout) για κάθε γραμμή του πίνακα. Κάθε στήλη του TableRow μπορεί να περιέχει ένα View (ή διάταξη με θυγατρικά αντικείμενα view). Μπορεί επίσης να καθοριστεί ο αριθμός στήλης (ξεκινώντας από το μηδέν) και να παρακαμφτούν στήλες που δεν θέλουμε, διαφορετικά το αντικείμενο view τοποθετείται στην επόμενη στήλη δεξιά. Το μέγεθος των στηλών ακολουθεί το μέγεθος του μεγαλύτερου View αυτής της στήλης.



Εικόνα 2.8: Παράδειγμα Table διάταξης στο Android

2.3.8.2 ListView

Μια List View είναι ένα ViewGroup το οποίο παρουσιάζει μια λίστα από scrollable αντικείμενα. Τα στοιχεία της λίστας παρέχονται από έναν Adapter ο οποίος τα αντλεί από την πηγή τους, τα μετατρέπει σε κατάλληλα view και τα τοποθετεί στη λίστα.



Εικόνα 2.9: Παράδειγμα List View διάταξης στο Android

2.3.8.3 Dialogs

Ένας διάλογος (dialog), είναι ένα μικρό παράθυρο το οποίο προτρέπει τον χρήστη να πάρει μια απόφαση ή να εισάγει επιπρόσθετες πληροφορίες. Ένας διάλογος δεν γεμίζει όλη την οθόνη και συνήθως χρησιμοποιείται όταν ο χρήστης πρέπει να κάνει κάποια ενέργεια για να μπορέσει να συνεχίσει. Όταν ο διάλογος εμφανίζεται, τότε η activity χάνει την εστίαση (focus) της και η εφαρμογή εστιάζει σε αυτόν. Έτσι, ο χρήστης μπορεί να αλληλεπιδράσει μόνο με τον διάλογο.

Τα δύο πιο βασικά είδη διαλόγου είναι:

Alert Dialog

Ένας διάλογος ο οποίος αποτελείται από ένα τίτλο, μερικά κουμπιά, ένα κείμενο ή μια λίστα επιλογών.

Progress Dialog

Ένας διάλογος προόδου. Χρησιμοποιείται για να δείξει στον χρήστη την πρόοδο μιας διαδικασίας. Όταν εκτελούνται χρονοβόρες διαδικασίες, όπως για παράδειγμα φόρτωση εικόνων ή σύνδεση με κάποιον εξυπηρετητή μέσω διαδικτύου, για να μην βλέπει ο χρήστης μια παγωμένη ή κενή οθόνη, χρησιμοποιείται το progress dialog για να ενημερώνει τον χρήστη για την πρόοδο της διαδικασίας.

Dialog Fragment

Ένα ακόμα ειδικό είδος διαλόγου είναι και τα Dialog Fragment, τα οποία ουσιαστικά είναι ένας διάλογος αλλά έχει ως βάση ένα Fragment και καλείται μέσα από την activity που το περιέχει. Στην παρακάτω Εικόνα 2.10 φαίνονται δύο παραδείγματα διαλόγων.



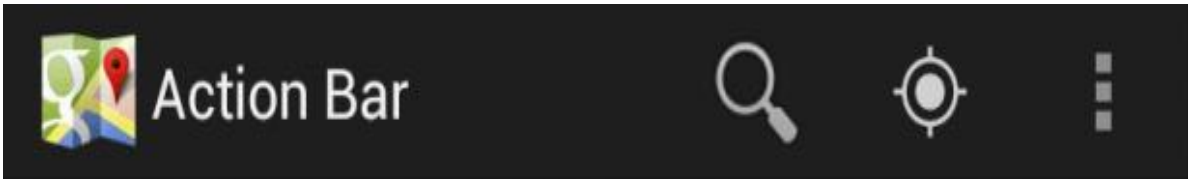
Εικόνα 2.10: Παράδειγμα dialogs στο Android

2.3.8.4 Action Bar

Η γραμμή δράσης είναι ένα χαρακτηριστικό παράθυρο που προσδιορίζει τη θέση του χρήστη, και παρέχει ενέργειες του χρήστη και τους τρόπους πλοήγησης. Χρησιμοποιώντας τη γραμμή δράσης, προσφέρεται στους χρήστες ένα οικείο περιβάλλον σε όλες τις εφαρμογές που το σύστημα προσαρμόζεται χάρη στις διαφορετικές διαμορφώσεις οθόνης.

Κάνει σημαντικές δράσεις εμφανέστερες και προσβάσιμες με έναν προβλέψιμο τρόπο (όπως Search). Υποστηρίζει συνεπή πλοήγηση και προβολή

μεταγωγής μέσα σε εφαρμογές (με καρτέλες ή drop-down λίστες). Προκειμένου όμως το Action bar να είναι λειτουργικό σε όλες τις εκδόσεις του Android θα πρέπει να ενσωματωθεί μία από τις δύο διαθέσιμες βιβλιοθήκες που είναι αυτή την στιγμή διαθέσιμες για τους προγραμματιστές. Η μία είναι της google ή v7 και η δεύτερη είναι η Action Bar Sher lock.



Εικόνα 2.11: Παράδειγμα Action Bar στο Android

2.3.8.5 Input Control

Σε αυτό το σημείο δεν θα μπορούσε να μην γίνει αναφορά στα γνωστά input control. Τα κύρια input control στο Android λίγο πολύ είναι τα ίδια με κάθε λειτουργικό σύστημα που περιέχει διεπιφάνεια χρήστη. Αναφορικά μερικά από αυτά είναι, edit text, spinner κ.α. Ουσιαστικά αυτά τα στοιχεία είναι που ενθυλακώνουν τα δεδομένα που εισάγονται και τα μεταφέρουν προς κάθε είδους επεξεργασία είτε αυτή είναι στο κινητό είτε σε κάποιον server.

2.4 Επίλογος

Συνοψίζοντας, έγινε μια προσπάθεια ο αναγνώστης να κατανοήσει τα συστατικά στοιχεία μίας εφαρμογής Android όπως Activity, Fragment, ActionBar, Lay outs κ.α, που είναι ο «πολύτιμος λίθος» στην ανάπτυξη Android εφαρμογών, προκειμένου να προχωρήσουμε στην αναλυτική περιγραφή τόσο της εφαρμογής όσο των τεχνολογιών και του προγραμματιστικού στυλ που ακολουθήθηκε.

3 Αρχιτεκτονική και τεχνολογίες που χρησιμοποιήθηκαν

3.1 Εισαγωγή

Στο παρόν κεφάλαιο θα περιγραφεί η εφαρμογή που αναπτύχθηκε για περιβάλλον Android. Θα αναλυθεί η εξωτερική δομή της, δηλαδή οι φάκελοι στους οποίους οργανώνεται, μαζί με τα αρχεία που την αποτελούν. Ακόμη θα αναλυθούν οι τεχνολογίες που χρησιμοποιήθηκαν καθώς και οι γλώσσες προγραμματισμού προκειμένου η εφαρμογή να γίνει πλήρως λειτουργική στο πρακτικό της μέρος και πλήρως κατανοητή από τους αναγνώστες.

3.2 Περιγραφή του Project

Ο τίτλος της εφαρμογής που θα αναπτυχθεί είναι «Προσφορά και ζήτηση αγοράς εργασίας». Με τον όρο "προσφορά" εκφράζεται γενικά σε κάποια δεδομένη χρονική στιγμή ή ακόμα και περίοδο η διαθέσιμη ή προσφερόμενη προς αγορά ποσότητα ενός οικονομικού αγαθού ή κάποιου συντελεστή παραγωγής. Ο όρος ζήτηση, όπως και ο όρος "προσφορά", εκφράζει επίσης ποσότητα, και συγκεκριμένα την ποσότητα των ζητούμενων αγαθών για κάλυψη αναγκών, (εμπορευμάτων, εργασίας κ.λ.π).

Ο σκοπός της εφαρμογής είναι να δοθεί στους χρήστες της η δυνατότητα να ζητούν αλλά και να προσφέρουν αγορά εργασίας με την χρήση της τεχνολογίας και συγκεκριμένα της κινητής τηλεφωνίας. Η εφαρμογή αναπτύχθηκε με επίκεντρο τον χρήστη και κατ' επέκταση την ευχρηστία και την απλή λειτουργικότητα της εφαρμογής ως προς αυτόν.

Πιο συγκεκριμένα ο χρήστης θα έχει την δυνατότητα αφού γίνει μέλος (register, Log In) να δημιουργεί το δικό του βιογραφικό και να ζητά εργασία τόσο με το βιογραφικό του όσο και προσαρμοσμένα ανάλογα με τις δικές του ανάγκες. Μερικές από αυτές τις ανάγκες μπορεί να είναι είτε ο μισθός, η γλώσσα, η χώρα ακόμα και η πόλη που θέλει να δουλέψει.

Στη συνέχεια, αφού ο χρήστης επιλέξει κάποια ελεύθερη θέση εργασίας του παρουσιάζονται τα αναλυτικά στοιχεία τις θέσεις (τηλέφωνο επικοινωνίας,

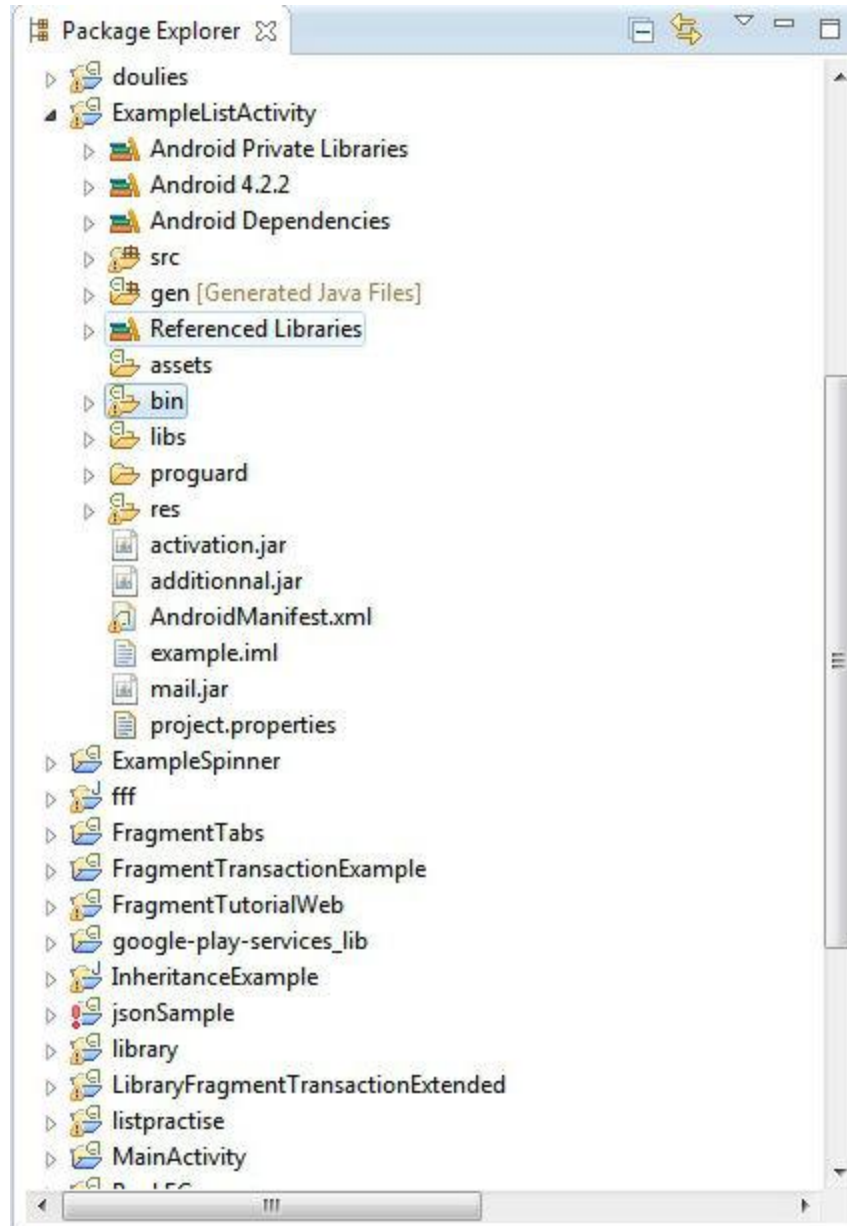
περιγραφή θέσης κ.α) και του δίνεται η δυνατότητα είτε να στείλει στον εργοδότη e-mail με το βιογραφικό του σημείωμα είτε να οδηγηθεί (google maps) στην διεύθυνση της εταιρίας.

Στην συνέχεια, ο χρήστης μπορεί ταυτόχρονα με τον ίδιο λογαριασμό μέλους να προσφέρει εργασία. Αφού πρώτα συμπληρώσει τα απαραίτητα παιδιά, η θέση εργασίας μπορεί να γίνει ορατή από όλους τους χρήστες που αναζητούν εργασία και εκτός αυτού ενημερώνονται εκείνοι που είναι πιο πιθανό να προσληφθούν στην συγκεκριμένη θέση.

Ο εργοδότης μπορεί να βλέπει ανά πάσα στιγμή τις θέσεις προσφοράς εργασίας που δημιούργησε στην εφαρμογή αλλά και να τις διαγράψει εφόσον το επιθυμεί ο ίδιος ή την διαγράψουμε εμείς μετά από κάποιο συγκεκριμένο χρονικό διάστημα.


3.3 Δομή του project στο Eclipse

Η δομή της παρούσας εφαρμογής όπως και κάθε εφαρμογής για Android, αποτελείται από φακέλους και διάφορα αρχεία σε μορφή δέντρου. Ένας προγραμματιστής Android πρέπει να τοποθετεί τις κλάσεις και τα αρχεία της εφαρμογής στους κατάλληλα σχεδιασμένους φακέλους που του δίνονται. Η συγκεκριμένη εφαρμογή αναπτύχθηκε στο περιβάλλον eclipse και η δομή της απεικονίζεται στην παρακάτω εικόνα:

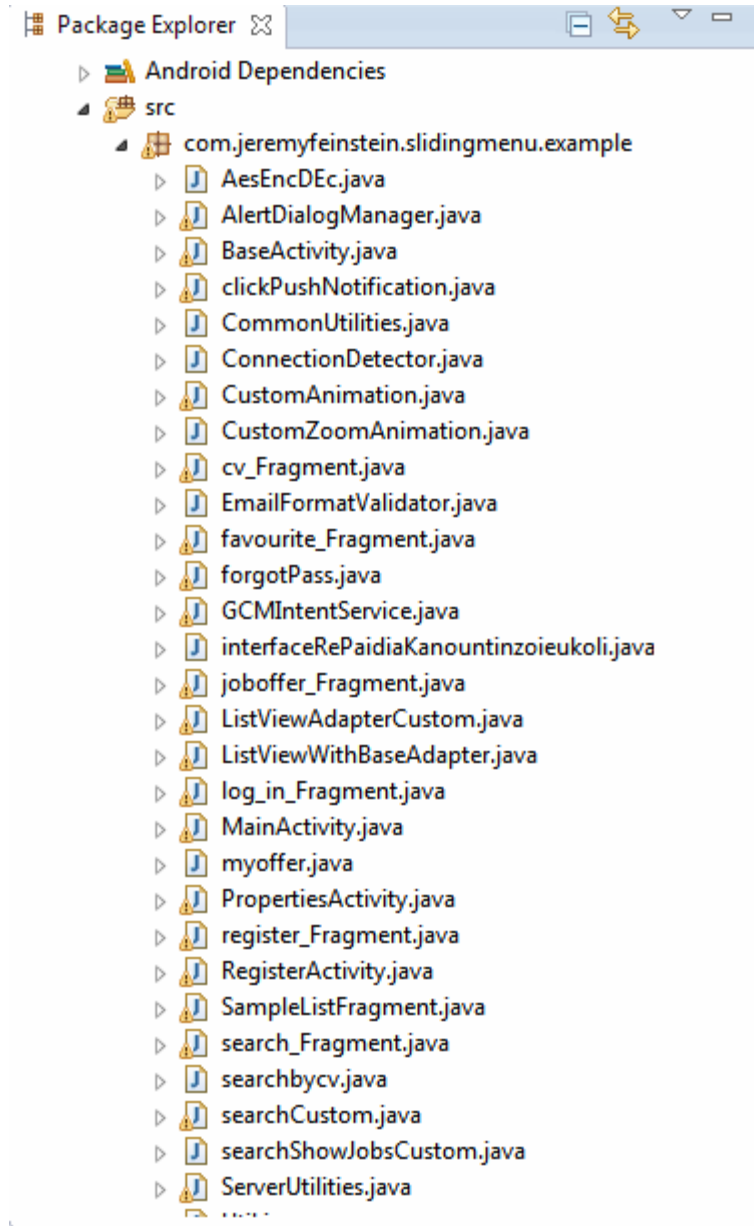


Εικόνα 3.1: Δενδρική δομή της εφαρμογής στο περιβάλλον Eclipse.

Συγκεκριμένα βλέπουμε:

 src/

Είναι ένας από τους βασικότερους φακέλους της εφαρμογής και αυτό γιατί περιέχει όλες τις κλάσεις και τα πακέτα (package) της εφαρμογής.



Εικόνα 3.2: Ξεδίπλωμα βασικού πακέτου εφαρμογής με της κλάσεις του.

Το πακέτο ***com.jeremyfinstein.slidingmenu.example*** σύμφωνα με την παραπάνω εικόνα (3.2) είναι όπως φαίνεται το πιο σημαντικό γιατί περιέχει σχεδόν όλη την λειτουργικότητα της εφαρμογής.

Android 4.2.2 , Android Private Libraries, Android dependencies

Οι 3 αυτοί φάκελοι περιέχουν βιβλιοθήκες που δημιουργούνται αυτόματα από το σύστημα, αλλά και άλλες βιβλιοθήκες που τοποθετήθηκαν από τον προγραμματιστή για την μέγιστη λειτουργικότητα της. Για παράδειγμα μερικές από τις βιβλιοθήκες που προστέθηκαν είναι :

- activation.jar, mail.jar προκειμένου ο χρήστης να έχει την δυνατότητα να στέλνει e-mail μέσα από την εφαρμογή.
- gcm.jar προκειμένου ο χρήστης να ενημερώνετε για προσφορές εργασίας που τον ενδιαφέρουν με την τεχνολογία push notification που θα αναλυθεί στην συνέχεια του κεφαλαίου.

assets

Στον φάκελο αυτό μπορεί να αποθηκευτεί οποιοδήποτε αρχείο. Βέβαια, τα Application Resources αποθηκεύονται σε ειδικό φάκελο που θα αναλυθεί στη συνέχεια με αποτέλεσμα τις περισσότερες φορές αυτός ο φάκελος να είναι άδειος.

bin

Ο φάκελος αυτός δημιουργείται από το σύστημα κατά τη δημιουργία του project και ανανεώνεται κάθε φορά που το project γίνεται build επιτυχώς και τρέχει. Μέσα σε αυτό το φάκελο, μετά από ένα επιτυχές build, μπορεί να βρεθεί και το εκτελέσιμο αρχείο .apk της εφαρμογής.

Android Manifest.xml

Τα έργα Android χρησιμοποιούν ένα ειδικό αρχείο ρύθμισης παραμέτρων, το οποίο ονομάζεται αρχείο manifest του Android και το οποίο καθορίζει τις ρυθμίσεις των παραμέτρων της εφαρμογής, ρυθμίσεις όπως το όνομα και η έκδοση της εφαρμογής, τα δικαιώματα που απαιτούνται για την εκτέλεση της εφαρμογής και από ποια στοιχεία αποτελείται η εφαρμογή.

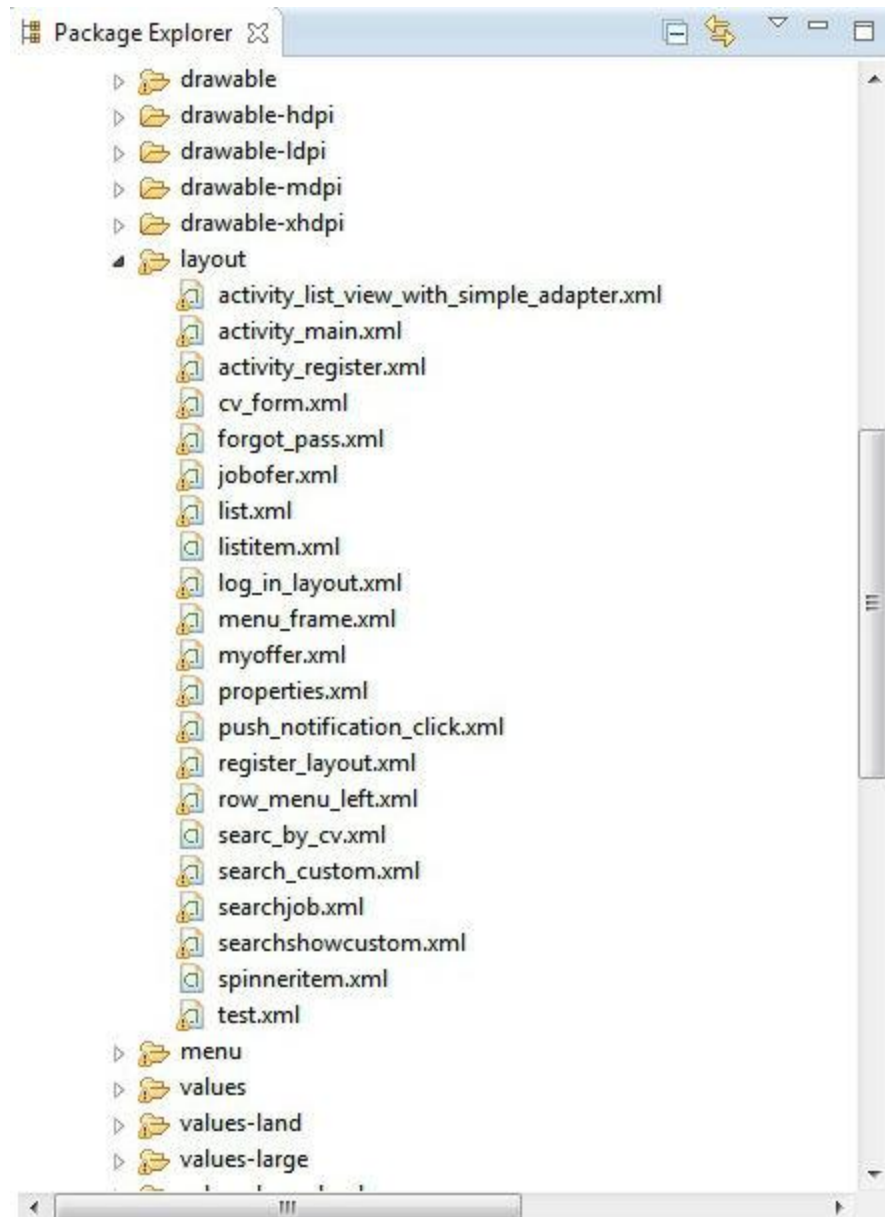
Το αρχείο manifest της εφαρμογής Android είναι ένα ειδικά μορφοποιημένο αρχείο XML, το οποίο πρέπει να συνοδεύει κάθε εφαρμογή Android.

Το αρχείο manifest του Android ονομάζεται **AndroidManifest.xml** και πρέπει να συμπεριλαμβάνεται στο κορυφαίο επίπεδο οποιουδήποτε έργου Android. Οι πληροφορίες σ' αυτό το αρχείο χρησιμοποιούνται απ' το σύστημα Android, προκειμένου να γίνουν οι εξής εργασίες:

1. Εγκατάσταση και αναβάθμιση του πακέτου της εφαρμογής.
2. Εμφάνιση λεπτομερειών της εφαρμογής, όπως το όνομά της, μία περιγραφή της και το εικονίδιο που θα χρησιμοποιούν οι χρήστες.
3. Προσδιορισμός απαιτήσεων συστήματος για την εφαρμογή, όπως ποια Android SDK υποστηρίζονται, τι ρυθμίσεις υλικού απαιτούνται (π.χ., ειδικές συσκευές πλοήγησης) και σε ποια χαρακτηριστικά πλατφόρμας βασίζεται η εφαρμογή (π.χ., χρησιμοποιεί στοιχεία πολλαπλής αφής).
4. Εκκίνηση δραστηριοτήτων εφαρμογής.
5. Διαχείριση δικαιωμάτων εφαρμογής.
6. ρύθμιση άλλων προχωρημένων παραμέτρων της εφαρμογής, όπως η λειτουργία ως υπηρεσία, δέκτης εκπομπής ή πάροχος υλικού.
7. Δυνατότητα καθορισμού ρυθμίσεων εφαρμογής, όπως η εξσφαλμάτωση και ρύθμιση παραμέτρων υλικού για δοκιμές.

 res/

Ο φάκελος στον οποίο αποθηκεύονται τα Application Resources



Εικόνα 3.3: Ξεδίπλωμα του φακέλου res/ της εφαρμογής .

Πιο αναλυτικά αυτός ο φάκελος περιέχει υποφακέλους όπου οι κυριότεροι εξ' αυτών είναι οι εξής:

- Drawable

Σε αυτόν τον φάκελο αποθηκεύονται όλα τα γραφικά της εφαρμογής. Με τον όρο γραφικά αναφερόμαστε σε κάθε τύπου εικόνας αλλά και σε αρχεία .xml που αφορούν τα γραφικά συστατικά της εφαρμογής.

- Layout

Σε αυτόν τον φάκελο αποθηκεύονται αρχεία .xml τα οποία σχετίζονται με το γραφικό περιβάλλον της εφαρμογής δηλαδή μέσα στα αρχεία αυτά ο προγραμματιστής γράφει τον κώδικα για τα layout στα οποία αναφερθήκαμε αναλυτικά στο προηγούμενο κεφάλαιο.

- Values

Στον φάκελο values αποθηκεύονται κατάλληλα αρχεία που περιέχουν ειδικές μεταβλητές που χρησιμοποιεί η εφαρμογή. Αντίστοιχα στους φακέλους values με το αντίστοιχο πρόσθετο (πχ values-sw600dp-land) περιέχονται οι μεταβλητές που χρησιμοποιεί το σύστημα σε αντίστοιχες περιπτώσεις (πχ συσκευή με smallest width 600dp και προσανατολισμός τοπίου).

- Anim

Στον φάκελο anim περιέχονται τα .xml αρχεία που περιγράφουν κάποια κίνηση animation που χρησιμοποιείται από την εφαρμογή.

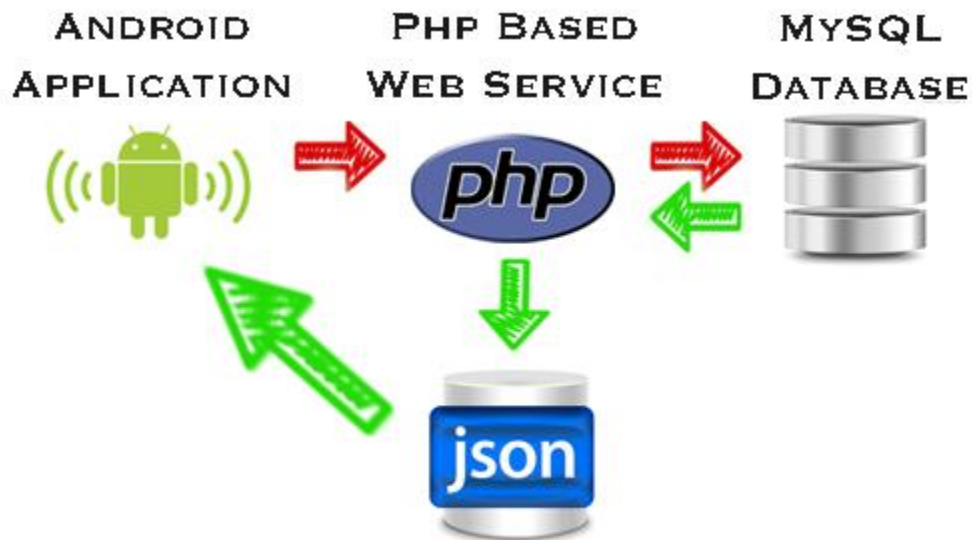
3.4 Τεχνολογίες και γλώσσες προγραμματισμού που χρησιμοποιήθηκαν

Σε αυτό το μέρος του κεφαλαίου θα ήταν εύλογο να αναπτυχθούν αναλυτικά τα περιβάλλοντα στα όποια αναπτύχθηκε η εφαρμογή, τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν καθώς και διάφορες τεχνολογίες που ήταν απαραίτητες για την ομαλή λειτουργία της. Πρώτα, όμως θα ήταν πιο κατανοητό να δοθεί μία γραφική αναπαράσταση της δομής του back end του application.

3.4.1 Γραφικός σχεδιασμός εφαρμογής

Στην εικόνα που ακολουθεί (Εικ:3.4) παρουσιάζεται αναλυτικά μια γραφική αναπαράσταση για την καλύτερη κατανόηση του back end της εφαρμογής. Πιο αναλυτικά χρησιμοποιείται μια MySQL βάση δεδομένων η οποία είναι εξωτερική (server). Στην βάση αυτή έχουν πρόσβαση όλες οι συσκευές Android με την προϋπόθεση να έχουν εγκατεστημένη την συγκεκριμένη εφαρμογή στο κινητό τους. Προκειμένου όμως να συνδεθεί η Android συσκευή με την MySQL γράφηκαν διάφορα αρχεία .php που παίζουν τον ρόλο του μεσάζοντα (Η ανάλυση τους θα γίνει στο επόμενο κεφάλαιο).

Επιπρόσθετα, για την επιστροφή διάφορων αποτελεσμάτων από την βάση στην συσκευή Android χρησιμοποιήθηκε Json τεχνολογία. Στην συνέχεια του κεφαλαίου αυτού θα αναλυθεί και θα εμπλουτιστεί η παρακάτω γραφική απεικόνιση με την κρυπτογράφηση και την GCM τεχνολογία η οποία είναι απαραίτητη προκειμένου να ενημερώνονται οι χρήστες για κάποια προσφορά εργασίας που τους ενδιαφέρει.



Εικόνα 3.4: Γραφική αναπαράσταση του Back end της εφαρμογής .

3.4.2 Γλώσσες προγραμματισμού

Σε αυτό το σημείο της πτυχιακής εργασίας θα αναφερθούν ποιες γλώσσες προγραμματισμού χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής με τίτλο “Προσφορά και ζήτηση αγοράς εργασίας”.

3.4.2.1 Java

Οι εφαρμογές στο Android γράφονται στη γλώσσα προγραμματισμού Java. Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε Playstation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά). Παρόλο που η εικονική μηχανή προσφέρει όλα αυτά (και όχι μόνο) τα πλεονεκτήματα, η Java αρχικά ήταν πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high-level) όπως η C και η C++. Εμπειρικές μετρήσεις στο παρελθόν έδειξαν ότι η C++ μπορούσε να είναι αρκετές φορές γρηγορότερη από την Java. Ωστόσο γίνονται προσπάθειες από τη Sun για τη βελτιστοποίηση της εικονικής μηχανής, ενώ υπάρχουν και άλλες υλοποιήσεις της εικονικής μηχανής από διάφορες εταιρίες (όπως της IBM), οι

οποίες μπορεί σε κάποια σημεία να προσφέρουν καλύτερα και σε κάποια άλλα χειρότερα αποτελέσματα. Επιπλέον με την καθιέρωση των μεταγλωττιστών JIT (Just In Time), οι οποίοι μετατρέπουν τον κώδικα byte απευθείας σε γλώσσα μηχανής, η διαφορά ταχύτητας από τη C++ έχει μικρύνει κατά πολύ.

Οι τελευταίες εκδόσεις του javac με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα.

Για να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο νί (γνωστός στο χώρο του Unix). Παρόλα αυτά, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (debugging). Υπάρχουν αρκετά διαθέσιμα, ενώ πολλά από αυτά έρχονται δωρεάν.

3.4.2.2 PHP

Η **PHP** είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Με άλλα λόγια έχει το ιδιαίτερο χαρακτηριστικό ότι ο κώδικάς της πρώτα μεταγλωττίζεται στον server και μετά φορτώνεται σαν ένα κανονικό html έγγραφο, χωρίς ο χρήστης να είναι σε θέση να δει τον αρχικό κώδικα. Με την τεχνολογία της Php μπορούμε να δημιουργήσουμε δυναμικές ιστοσελίδες (dynamic web pages).

Ανταγωνιστικές της τεχνολογίας Php είναι η τεχνολογία ASP (Server Pages) της εταιρείας Microsoft καθώς και η τεχνολογία CFML (ColdFusion Markup Language). Θα δούμε ένα πολύ απλό αρχικό παράδειγμα.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να

είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

3.4.2.2.1 Γιατί PHP

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε η γλώσσα προγραμματισμού php γιατί αν αναλογιστεί κανείς τα πλεονεκτήματα που έχει σε σχέση με τους ανταγωνιστές της όπως η ASP είναι πολλά. Μερικά από τα προτερήματα της είναι τα ακόλουθα:

- ✚ Υψηλή απόδοση.
- ✚ Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων.
- ✚ Ενσωματωμένες βιβλιοθήκες για πολλές web διεργασίες.
- ✚ Χαμηλό κόστος σε σχέση με την ASP η οποία δεν είναι open source.
- ✚ Ευκολία μάθησης και χρήσης.
- ✚ Μεταφερσιμότητα
- ✚ Διαθεσιμότητα του κώδικα προέλευσης.

3.4.2.3 XML

Για τον προγραμματισμό σε λειτουργικό Android είναι απαραίτητη η καλή γνώση XML. Όπως αναλύθηκε παραπάνω, στην δομή μιας Android εφαρμογής σχεδόν όλοι οι φάκελοι δέχονται μόνο αρχεία XML. Πιο αναλυτικά η Η XML (αγγλ. από το **Extensible Markup Language**) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (Word Wide Web Consortium), αλλά και σε

διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML. Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org και του iWork της εταιρίας Apple.

3.4.2.3.1 Βασική ορολογία

Το περιεχόμενο αυτής της ενότητας, βασίζεται στην προδιαγραφή XML 1.0. Δεν αποτελεί μία πλήρη λίστα όλων των όρων που υπάρχουν στη γλώσσα XML. Είναι μία εισαγωγή στα βασικά στοιχεία, που συναντώνται στην ανάπτυξη Android εφαρμογών.

Χαρακτήρας Unicode

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Επεξεργαστής και εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας

επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονομάζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

✚ Σήμανση και περιεχόμενο

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";" . Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

✚ Ετικέτα

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

✚ Στοιχείο

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

✚ Χαρακτηριστικό

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο

παράδειγμα παρακάτω, το στοιχείο **img** έχει δύο χαρακτηριστικά, τα **src** και **alt**: ``. Ένα άλλο παράδειγμα θα ήταν το `<step number="3">Connect A to B.</step>`, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

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

3.4.2.3.2 Παράδειγμα

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που κάνει χρήση όλων των παραπάνω εννοιών και στοιχείων.

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

3.4.2.4 MySQL

Η βάση δεδομένων MySQL έχει γίνει η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα λόγω της αυξημένης απόδοσης, υψηλής αξιοπιστίας και ευκολίας στην διαχείριση της. Η MySQL θεωρείται μια από τις πιο αξιόπιστες πλατφόρμες για την συγκέντρωση δεδομένων και για αυτό έχει κερδίσει την εκτίμηση τόσο των administrators όσο και των programmers. Διατίθεται εντελώς δωρεάν και υποστηρίζει εφαρμογές από τον τομέα των τηλεπικοινωνιών, ηλεκτρονικών καταστημάτων στο διαδίκτυο ως και συστήματα εξυπηρέτησης πελατών και μικροσυσκευές όπως PDAs. Χρησιμοποιείται σε περισσότερες από 6 εκατομμύρια εγκαταστάσεις κλιμακούμενες από μεγάλες εταιρίες μέχρι εξειδικευμένες εφαρμογές με ενσωματωμένες βάσεις δεδομένων σε όλο τον κόσμο.

Η MySQL καθίσταται η απόλυτη επιλογή μιας νέας γενιάς εφαρμογών που υλοποιούνται σε περιβάλλον LAMP (Linux. Apache. MySQL. PHP / Perl / Python).

Επίσης, η MySQL λειτουργεί σε περισσότερες από 20 πλατφόρμες συμπεριλαμβανομένων των Linux, Windows, OS/390, HP-UX, AIX, Netware. Προϋπόθεση για να έχουμε και να λειτουργούμε MySQL βάσεις δεδομένων είναι να μας παρέχετε ένας MySQL server. Η MySQL είναι το λογισμικό του διακομιστή βάσεων δεδομένων (database server software) που χρησιμοποιούμε.

Χρήσιμα εργαλεία είναι ο MySQL Administrator που έχει σχεδιαστεί για την διαχείριση ενός MySQL εξυπηρετητή και ο MySQL Query Browser που έχει σχεδιαστεί για να μας βοηθήσει να θέτουμε αιτήματα και να αναλύουμε δεδομένα, τα οποία είναι αποθηκευμένα στην MySQL βάση δεδομένων μας.

Η MySQL ανήκει στην γενική κατηγορία των DBMS (Database Management Systems) που έχουν σαν πρωταρχικό τους ρόλο την αποθήκευση των δεδομένων, την ελεγχόμενη και ασφαλή πρόσβαση στις πληροφορίες, και την διαχείριση των δικαιωμάτων με τα οποία οι χρήστες μπορούν να επέμβουν και να αλλάξουν στοιχεία πληροφοριών. Κάθε φορά που ένας χρήστης προσπαθεί να διαχειριστεί ένα όγκο δεδομένων προσθέτοντας νέα στοιχεία ή αφαιρώντας κάποια άλλα που δεν ισχύουν πια, το DBMS είναι υπεύθυνο να ακολουθήσει αυτή την διαδικασία από την αρχή της ενεργοποίησης της μέχρι το τέλος της.

3.4.2.4.1 Χαρακτηριστικά-Πλεονεκτήματα

Στην ανάπτυξη της συγκεκριμένης εφαρμογής Android χρησιμοποιήθηκε η MySQL μεταξύ άλλων βάσεων δεδομένων για τους εξής λόγους:

Επεκτασιμότητα και ευελιξία

Τα συστήματα της Sun για MySQL μπορούν να επεκταθούν, για να ικανοποιήσουν την εκθετική αύξηση του όγκου δεδομένων που χαρακτηρίζει τα εμπλουτισμένα πολυμέσα, τις Διαδικτυακές κοινότητες και άλλες υπηρεσίες Web. Εκτελείται οτιδήποτε από βαθιά ενσωματωμένες εφαρμογές με καταλαμβανόμενο χώρο μόλις 1MB ή μαζικές "αποθήκες" δεδομένων - με terabyte πληροφοριών. Επίσης τα μοναδικά στο είδος τους συστήματα της Sun μπορούν να "επεκταθούν μέσα στο διακομιστή, ενώ η τεχνολογία εικονικοποίησης της Sun μπορεί να

μειώσει τις απαιτήσεις σε κατανάλωση ρεύματος και χώρο και να επιτύχει εξοικονόμηση κόστους και μεγαλύτερο σεβασμό προς το περιβάλλον.

✚ Υψηλή διαθεσιμότητα

Εκτέλεση διαμορφώσεων υψηλής ταχύτητας κύριας αναπαραγωγής βασισμένη σε γραμμές και υβριδική αναπαραγωγή. Οι εξειδικευμένοι διακομιστές συμπλεγμάτων προσφέρουν άμεση μεταγωγή μετά από αστοχία.

✚ Πλεονεκτήματα αποθήκευσης δεδομένων και web.

Υψηλής απόδοσης μηχανισμός ερωτημάτων. Εξαιρετικά γρήγορη δυνατότητα εισαγωγής δεδομένων. Ισχυρή υποστήριξη για εξειδικευμένες λειτουργίες web - συμπεριλαμβανομένων των γρήγορων αναζητήσεων πλήρους κειμένου.

✚ Ισχυρή προστασία δεδομένων

Πανίσχυροι μηχανισμοί διασφάλισης της προσπέλασης μόνο από εξουσιοδοτημένους χρήστες. Υποστήριξη SSH και SSL για ασφαλείς και σίγουρες συνδέσεις. Ισχυρή κρυπτογράφηση δεδομένων και λειτουργίες αποκρυπτογράφησης.

✚ Ολοκληρωμένη ανάπτυξη εφαρμογή

Υποστήριξη για αποθηκευμένες διαδικασίες, κανόνες ενεργοποίησης, λειτουργίες, προβολές, δρομείς, γλώσσες SQL προτύπου ANSI, και πολλά περισσότερα. Βιβλιοθήκες προσθέτων για την ενσωμάτωση της υποστήριξης βάσεων δεδομένων MySQL σχεδόν σε κάθε εφαρμογή.

✚ Ευκολία διαχείρισης

Υπάρχει χρήση του προγραμματισμού συμβάντων - αυτόματος προγραμματισμός συνήθων επαναλαμβανόμενων εργασιών βασισμένων σε SQL για εκτέλεση σε διακομιστή βάσης δεδομένων. Επίσης ο μέσος χρόνος από τη λήψη λογισμικού μέχρι την ολοκλήρωση της εγκατάστασης είναι λιγότερος από δεκαπέντε λεπτά.

- ✚ Ελευθερία, ανοικτού πηγαίου κώδικα και υποστήριξη 24 ώρες το εικοσιτετράωρο, 7 ημέρες την εβδομάδα.

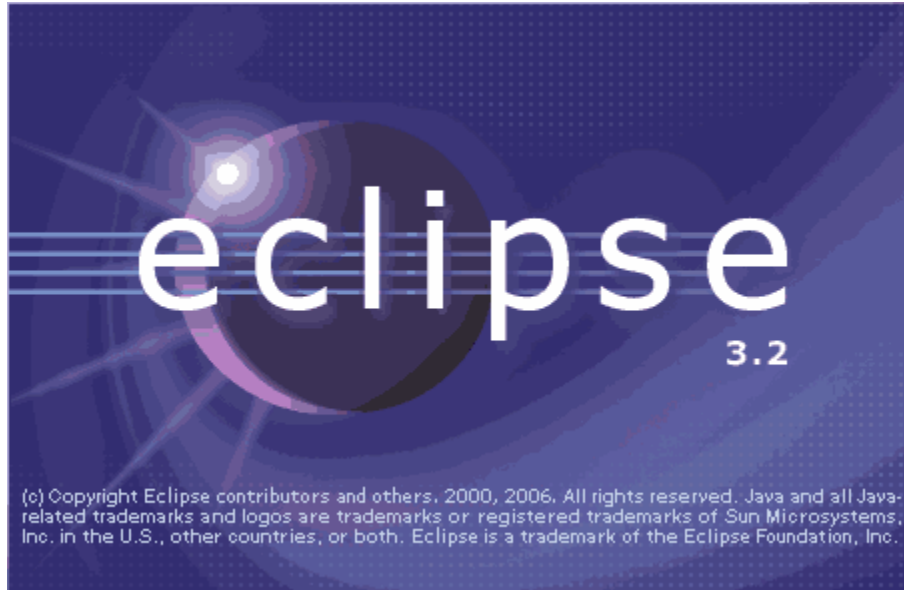
Μπορεί να προσφέρει εικοσιτετράωρη υποστήριξη και διαθέσιμη επανόρθωση μέσω του δικτύου MySQL. Εταιρική ποιότητα και έτοιμο για εταιρική χρήση από την εγκατάσταση ως την υποστήριξη.

3.4.3 Προγραμματιστικά περιβάλλοντα

Σε αυτό το σημείο θα αναλυθούν τα προγραμματιστικά περιβάλλοντα που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής καθώς και κάποιες απαραίτητες οδηγίες χρήσης.

3.4.3.1 Eclipse

Το Eclipse είναι ένα από τα δημοφιλέστερα IDE (Integrated Development Environment ή στα ελληνικά: Ολοκληρωμένο Περιβάλλον Ανάπτυξης) που χρησιμοποιείται από χιλιάδες προγραμματιστές παγκοσμίως για τη συγγραφή και εκτέλεση κώδικα. Η επιτυχία του οφείλεται στο λιτό του περιβάλλον το οποίο είναι φιλικό ακόμα και σε κάποιον αρχάριο προγραμματιστή, επίσης είναι σχεδιασμένο να λειτουργεί σε πολλά λειτουργικά συστήματα (Linux, Mac, Windows), αλλά κυρίως επειδή υποστηρίζει πολλές γλώσσες προγραμματισμού, από Java μέχρι C, C++, Perl, PHP, Python δίνοντας έτσι στο προγραμματιστή το ίδιο περιβάλλον εργασίας για τελείως διαφορετικούς κόσμους.



Εικόνα 3.5: Logo προγραμματιστικού περιβάλλοντος Eclipse.

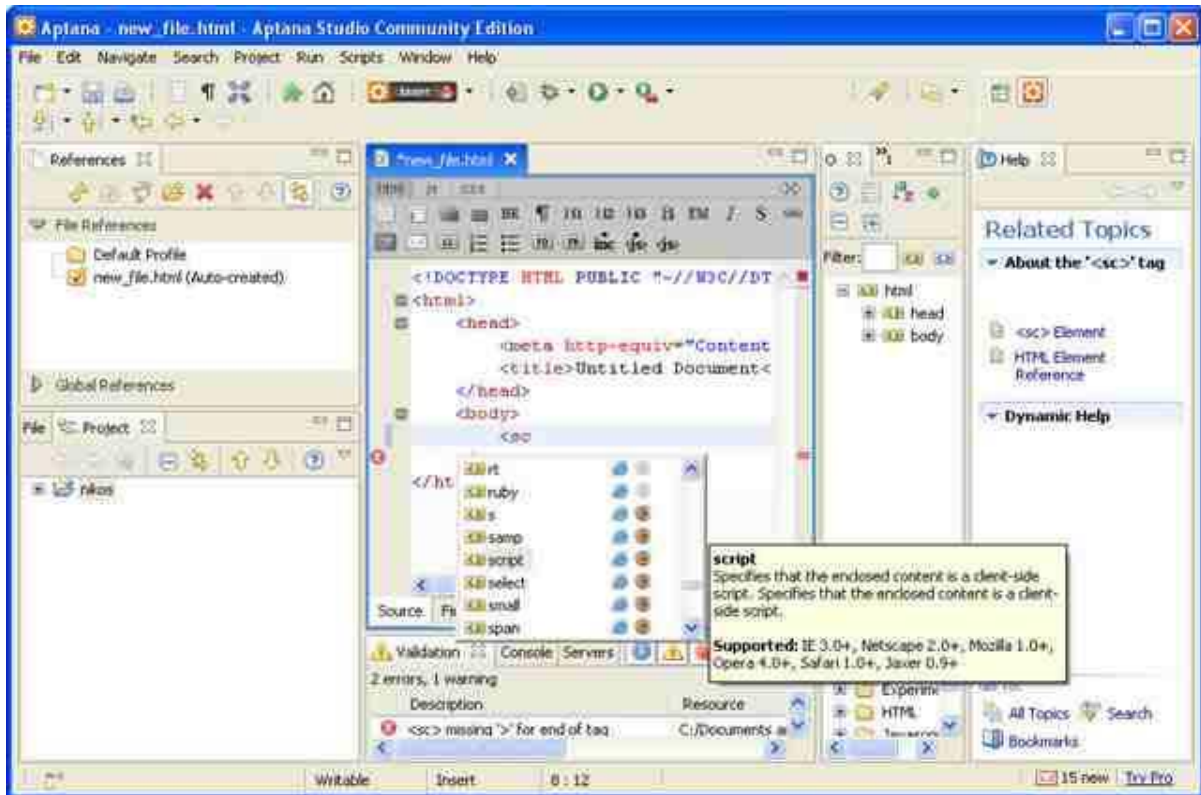
3.4.3.1.1 Προσαρμογή του Android στο eclipse

Το πρώτο πράγμα που πρέπει να εφαρμοστεί προκειμένου να έχουμε την δυνατότητα προγραμματισμού σε λειτουργικό Android είναι η ενσωμάτωση του Android SDK στο Eclipse. Αυτό μπορεί να γίνει με δύο τρόπους:

- ✚ Ο πρώτος είναι να κατεβάσουμε το sdk από την official σελίδα του Android(<http://developer.android.com/sdk/index.html>) και να γίνει import στο Eclipse όπως γίνεται import μία απλή βιβλιοθήκη.
- ✚ Ο δεύτερος είναι να κατεβάσουμε κατευθείαν το eclipse με το ενσωματωμένο sdk για Android (<http://developer.android.com/sdk.html>)

3.4.3.2 Aptana

Το Aptana Studio (www.apтана.com) αποτελεί ίσως την καλύτερη δωρεάν λύση για την κατασκευή html ιστοσελίδων με χρήση CSS, Javascript, DOM, PHP, Ajax κα. Οι δυνατότητές του για CSS, Javascript, DOM και Ajax συναγωνίζονται ισάξια ακόμη και το επαγγελματικό Dreamweaver. Η αυτόματη συμπλήρωση εντολών-tags και η αυτόματη μορφοποίηση είναι υποδειγματικές. Έχει δυνατότητες προεπισκόπησης τόσο για IE όσο και για Firefox. Η προεπισκόπηση για Firefox εκτελεί και Javascript σενάρια ενώ π.χ. στα δωρεάν λογισμικά NVU και Komprozer πρέπει να ανοίξουμε το αρχείο με τον browser. Ένα στιγμιότυπο του Aptana Studio είναι αυτό στην εικόνα που ακολουθεί:

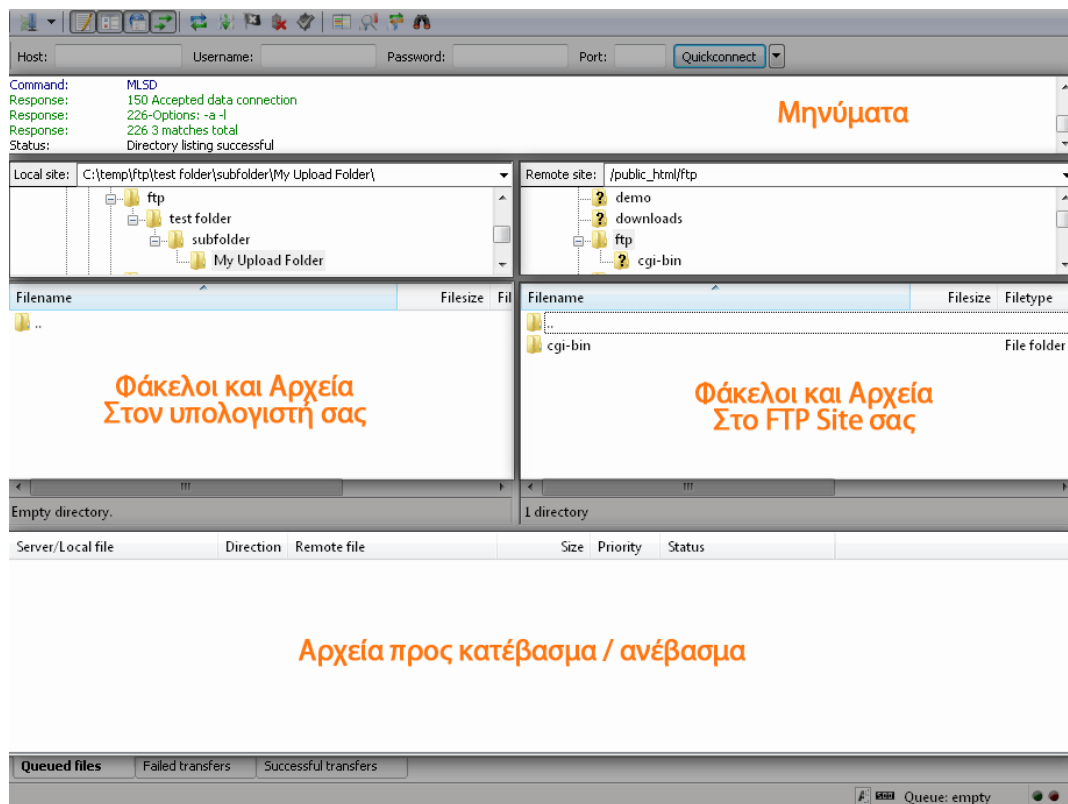


Εικόνα 3.6: Στιγμιότυπο Aptana Studio

3.4.3.3 Filezilla

FileZilla ονομάζεται ένας FTP client (Το FTP είναι το πρωτόκολλο που επιτρέπει την διαχείριση αρχείων στο Internet) που μπορούμε να χρησιμοποιήσουμε για να κατεβάζουμε τα αρχεία της ιστοσελίδας στον server μας. Υπάρχουν διαθέσιμες εκδόσεις για Windows, Linux και Mac OS X. Το FileZilla είναι ένα ελεύθερο λογισμικό και μπορεί να το κατεβάσει ο καθένας από το ακόλουθο Link (<http://filezilla-project.org/download.php?type=client>).

Ο χειρισμός του FileZilla είναι απλός και πολύτιμος στην ανάπτυξη επαγγελματικών διαδικτυακών και μη εφαρμογών. Η κύρια οθόνη που αξίζει να αναλυθεί στα πλαίσια αυτής της πτυχιακής εργασίας είναι η παρακάτω Εικ:3.7:



Εικόνα 3.7: FileZilla ανάλυση κεντρικής οθόνης

Σε αυτήν την οθόνη του FileZilla (Εικόνα 3.7) και συγκεκριμένα στο επάνω μέρος βλέπουμε τα Edit Text (Host, User Name, Password) στα οποία πληκτρολογούνται τα στοιχεία του server μας προκειμένου να γίνει η σύνδεση με την MySql βάση δεδομένων. Στην συνέχεια βλέπουμε τους φακέλους και τα αρχεία

που βρίσκονται στον σκληρό δίσκο του ηλεκτρονικού υπολογιστή και ακριβώς δίπλα βλέπουμε τα αρχεία που είναι ήδη στον server. Η μετακίνηση αρχείων από τον τοπικό υπολογιστή μας στον server γίνεται με ένα απλό drag and drop από το ένα παράθυρο στο άλλο.

3.4.3.4 PhpMyAdmin

Το εργαλείο phpMyAdmin παρέχει εκτεταμένη διαχείριση βάσεων δεδομένων MySQL στο Διαδίκτυο. Υποστηρίζει μια ευρεία λίστα λειτουργιών μέσω γραφικής διεπαφής, όπως σχετικά με πίνακες, πεδία, συσχετίσεις, ευρετήρια, χρήστες και δικαιώματα. Παράλληλα, είναι δυνατή και η εκτέλεση οποιασδήποτε εντολής **SQL**. Είναι ανοιχτού κώδικα, αν και σε πολλούς διαδικτυακούς χώρους μπορεί κάνεις να το βρει και να το εγκαταστήσει εντελώς δωρεάν, ο επίσημος διαδικτυακός χώρος είναι http://www.phpmyadmin.net/home_page/index.php. Παρακάτω, θα σας παραθέσω μία εικόνα (3.7) με το βασικό παράθυρο του PhpMyAdmin και ακολουθούν τα χαρακτηριστικά του.



Εικόνα 3.7: Κεντρική οθόνη phpMyAdmin

3.4.3.4.1 Χαρακτηριστικά

Ο σκοπός της phpMyAdmin είναι να δώσει στο χρήστη μια ολοκληρωμένη διαχείριση βάσεων δεδομένων. Παρακάτω αναφέρονται κάποια από τα κύρια χαρακτηριστικά λειτουργικότητας του phpMyAdmin τα οποία χρησιμοποιήθηκαν στο συγκεκριμένο έργο.

Τα βασικά χαρακτηριστικά περιλαμβάνουν:

1. Δημιουργία βάσης δεδομένων, διαγραφή, μετονομασία
2. Πίνακας δημιουργία, μετονομασία, αντιγραφή, διαγραφή
3. Πίνακας συντήρηση δομή, συμπεριλαμβανομένων των δεικτών
4. Ειδικών δράσεων πίνακα (επισκευή, βελτιστοποίηση, αλλάζοντας τον τύπο)
5. Δεδομένα εισαγωγής, τροποποίηση, διαγραφή
6. Τα δεδομένα εμφανίζονται στην οριζόντια και κάθετη λειτουργία, και εκτύπωσης
7. Δυναμικά δεδομένα uploading
8. Αναζήτηση δεδομένων (πίνακας ή βάση δεδομένων)
9. Παρτίδα-φόρτωση των δεδομένων (εισαγωγή)
10. Εξάγει δεδομένα σε διάφορες μορφές, με συμπίεση

3.4.4 Τεχνολογίες που χρησιμοποιήθηκαν

3.4.4.1 JSON

Σε αυτό το σημείο είναι εύλογο να αναλυθεί εις βάθος η γλώσσα προγραμματισμού ανταλλαγής δεδομένων (JSON). Στη συγκεκριμένη πτυχιακή χρησιμοποιείται για να μας επιστρέφει οποιοδήποτε δεδομένο είτε από τον server είτε από τον server της google GCM (θα αναλυθεί στην συνέχεια του κεφαλαίου) πράγμα που την κάνει ιδιαίτερα σπουδαία, πιο αναλυτικά:

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν και γράψουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος

1999. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

- ✚ Μια συλλογή από ζευγάρια ονομάτων/τιμών.

Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.

- ✚ Μία ταξινομημένη λίστα τιμών.

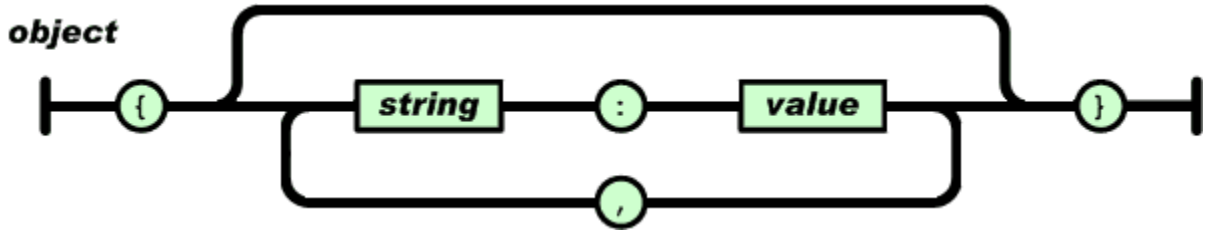
Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Αυτά είναι τα universal data structures. Ουσιαστικά όλες οι μοντέρνες γλώσσες προγραμματισμού τα υποστηρίζουν με τον έναν ή τον άλλον τρόπο. Λογικό είναι πως ένα πρότυπο δεδομένων το οποίο είναι εύκολα μεταβαλλόμενο με γλώσσες προγραμματισμού οι οποίες επίσης είναι βασισμένες σε αυτές τις δομές.

3.4.4.1.1 JSON μορφές

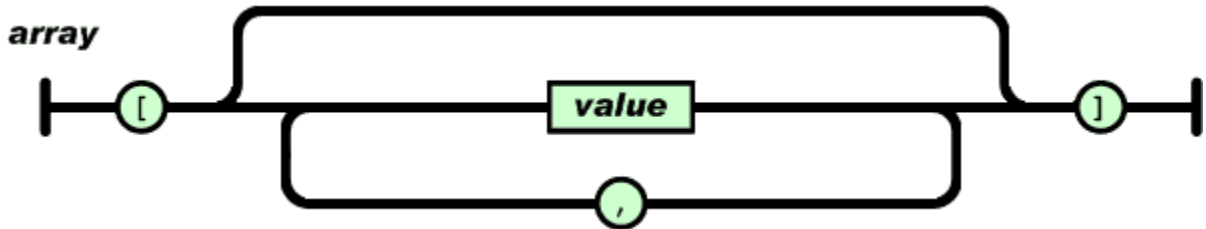
Ένα αντικείμενο (object) είναι ένα άτακτο σύνολο από ζευγάρια ονομάτων/τιμών.

Ένα αντικείμενο (object) ξεκινάει με { (αριστερό άγκιστρο) και τελειώνει με } (δεξιό άγκιστρο). Κάθε όνομα ακολουθείται από : (άνω-κάτω τελεία) και τα ζευγάρια ονόματος/τιμής χωρίζονται από , (κόμμα).



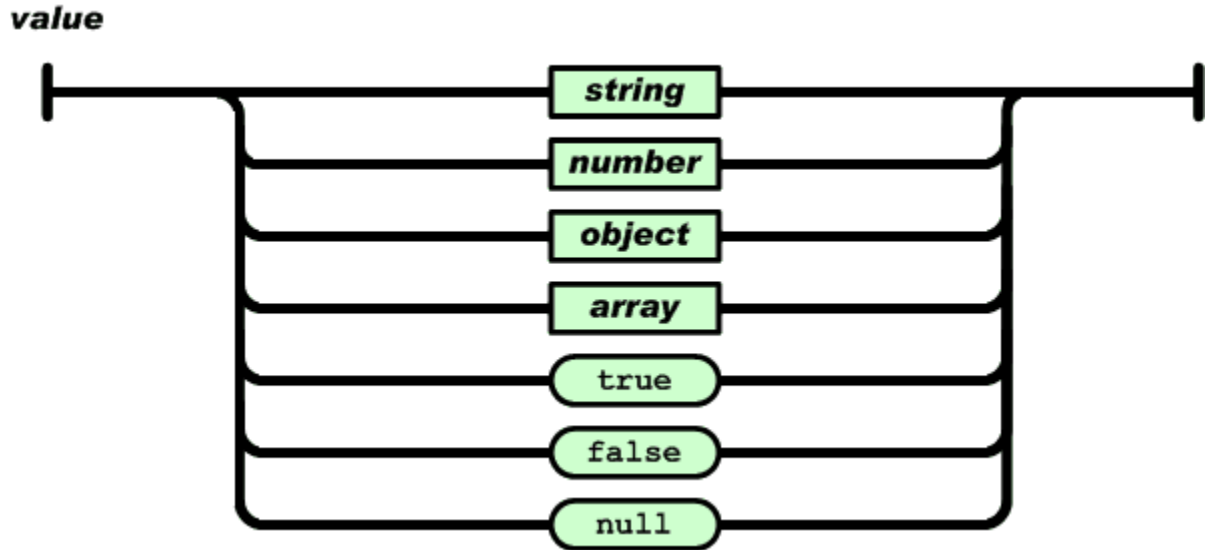
Εικόνα 3.8 Json Object

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



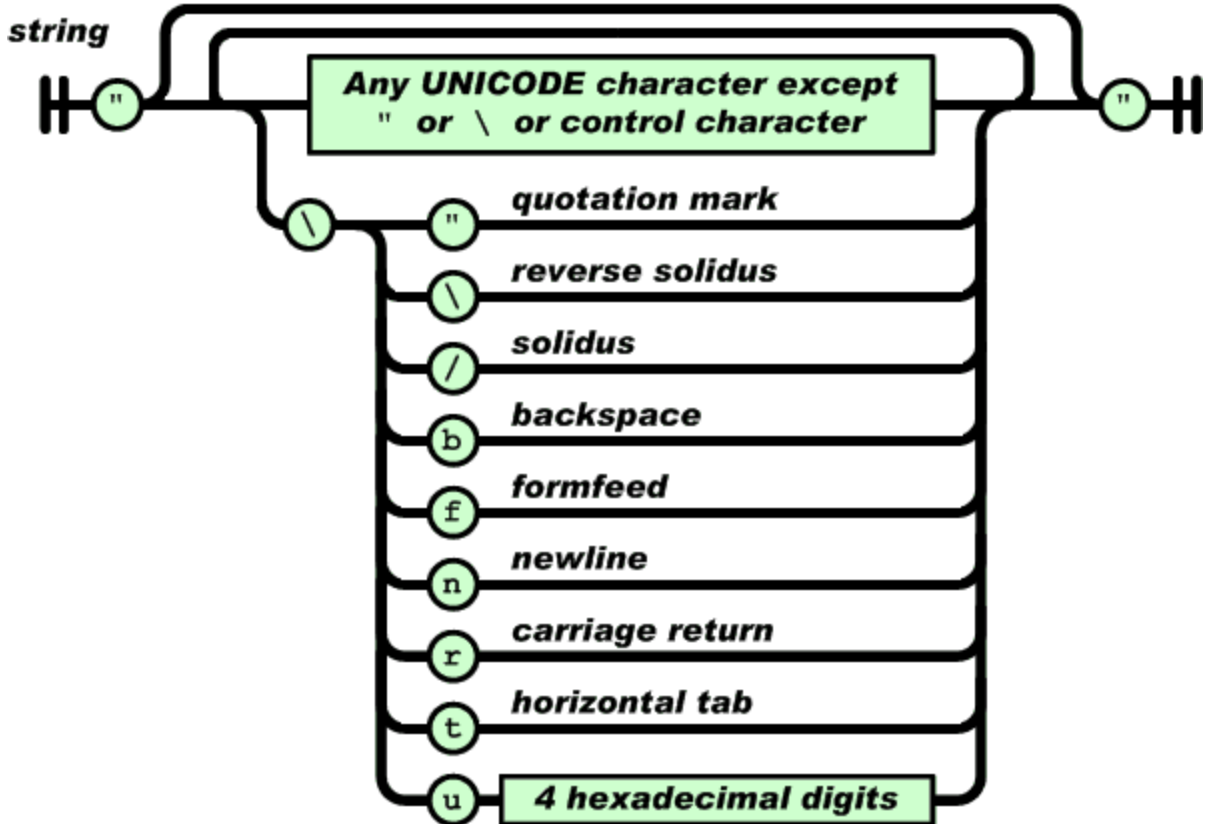
Εικόνα 3.9 Json Array

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



Εικόνα 3.10 Json Value

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



Εικόνα 3.11 Json String

3.4.4.1.2 Παράδειγμα

Ένα παράδειγμα Json μορφής είναι το εξής :

```
{
  "players": [
    {
      "id": 863,
      "name": "ΔΗΜΗΤΡΗΣ",
      "surname": "ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΣ"
    },
    {
      "id": 131,
      "name": "ΝΙΚΟΣ",
      "surname": "ΓΕΩΡΓΕΑΣ"
    },
    {
      "id": 101,
      "name": "ΠΑΝΤΕΛΗΣ",
      "surname": "ΚΑΦΕΣ"
    },
    {
      "id": 124,
      "name": "ΚΩΣΤΑΣ",

```

```
    "surname": "ΜΑΝΩΛΑΣ"  
  },  
  {  
    "id": 115,  
    "name": "ΝΙΚΟΣ",  
    "surname": "ΚΑΡΑΜΠΕΛΑΣ"  
  },  
  {  
    "id": 297,  
    "name": "ΜΑΥΡΟΥΔΗΣ",  
    "surname": "ΜΠΟΥΓΑΙΔΗΣ"  
  },  
  {  
    "id": 114,  
    "name": "ΓΡΗΓΟΡΗΣ",  
    "surname": "ΜΑΚΟΣ"  
  },  
  {  
    "id": 109,  
    "name": "ΠΕΡΕΙΡΑ",  
    "surname": "ΛΕΟΝΑΡΝΤΟ"  
  }  
]  
}
```

3.4.4.1.3 *Json Parser*

Αφού αναλύσαμε διεξοδικά την Json τεχνολογία δημιουργείτε το εύλογο ερώτημα πώς θα διαβαστεί το Json που επιστρέφεται μετά από κάποια επεξεργασία των δεδομένων στον server από το Android και κατά επέκταση από την Java? Για αυτόν ακριβώς το λόγο δημιουργήθηκε ο Parser που εξυπηρετεί ακριβώς αυτόν τον σκοπό.

3.4.4.2 **GCM (Google Cloud Message)**

Το Google Cloud Messaging (GCM) είναι μια δωρεάν υπηρεσία που μας διευκολύνει να στείλουμε δεδομένα από τους διακομιστές στις εφαρμογές Android. Αυτό θα μπορούσε να είναι ένα μήνυμα που υποδεικνύει στην εφαρμογή Android ότι υπάρχουν νέα δεδομένα προς ανάκτηση από το διακομιστή (για παράδειγμα, μια ταινία που ανέβασε ένας φίλος) ή θα μπορούσε να είναι ένα μήνυμα που να περιέχει έως και 4 KB ωφέλιμου φορτίου δεδομένων (επομένως οι εφαρμογές όπως η άμεση ανταλλαγή μηνυμάτων μπορούν να καταναλώσουν άμεσα το μήνυμα).

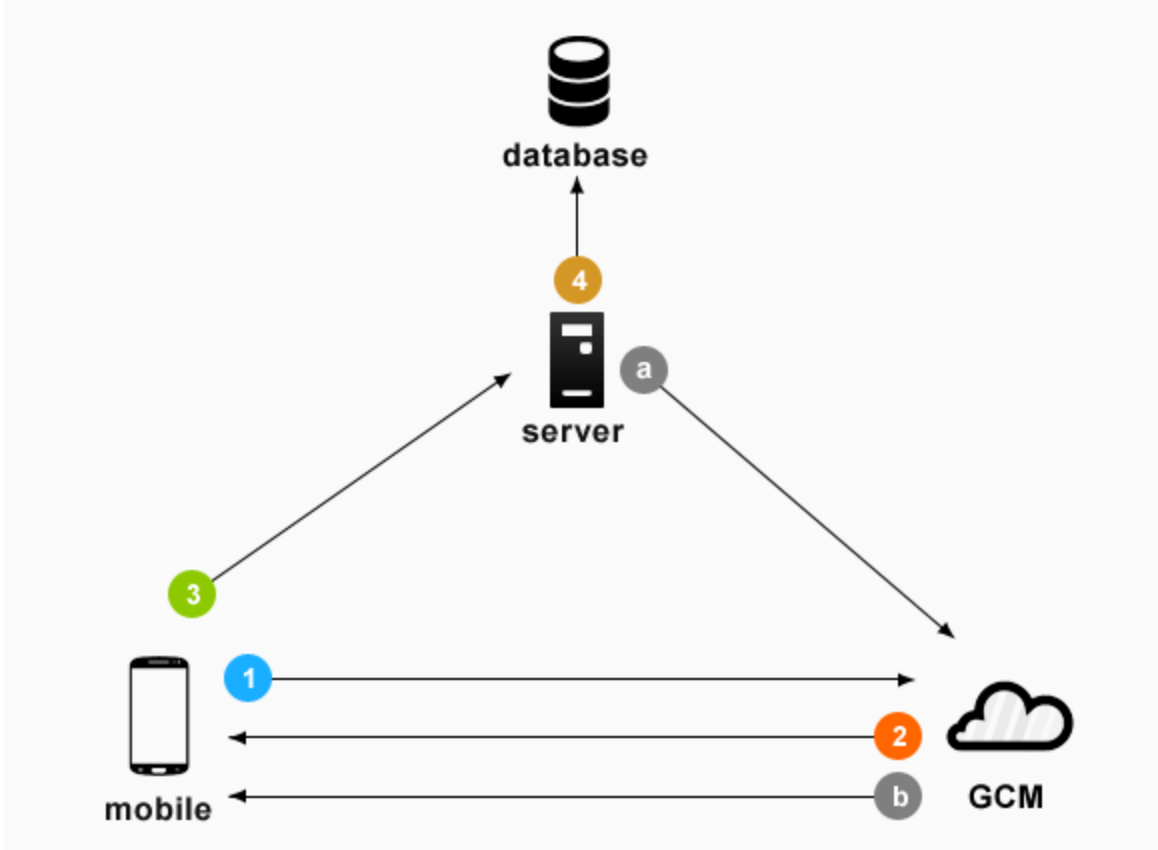
3.4.4.2.1 Γιατί Google Cloud Message

Το Google Cloud Messaging (GCM) είναι ότι πιο πρόσφατο υπάρχει αυτήν την στιγμή στην ανάπτυξη Android εφαρμογών. Στην συγκεκριμένη εφαρμογή χρησιμοποιείτε στο να ενημερώνονται οι χρήστες για νέες προσφορές εργασίας που τους ενδιαφέρει. Ένας εναλλακτικός τρόπος στο να επιτυγχάναμε το ίδιο αποτέλεσμα θα ήταν να φτιάχναμε ένα background service στο κινητό το οποίο να το προγραμματίζαμε να «κοιτάει» ανά τακτά χρονικά διαστήματα την βάση δεδομένων μας. Σίγουρα η παραπάνω διαδικασία δεν είναι μια καλή πρακτική επίλυσης του προβλήματος αφού έτσι περιορίζουμε τον χρήστη μας να έχει ένα πολύ εξελιγμένο κινητό καθώς η παραπάνω διαδικασία απαιτεί επεξεργαστική ισχύει και δέσμευση μνήμης ram.

Απεναντίας, αφήνουμε όλη αυτήν την λειτουργία στο να την κάνει ο cloud server της google. Πρακτικά αυτό σημαίνει ότι αν υπάρχει μία νέα καταχώρηση προσφοράς εργασίας ο χρήστης ενημερώνεται χωρίς καμία επεξεργαστική λειτουργία του κινητού του τηλεφώνου. Στη συνέχεια αναλύεται η παραπάνω διαδικασία.

3.4.4.2.2 Δομή

Πριν αναλυθεί η διαδικασία για το πώς λειτουργεί και ενσωματώνεται το GCM στην εφαρμογή μας, θα ήταν σκόπιμο να παρατηρηθεί η γραφική αναπαράσταση αυτής της διαδικασίας στην εικόνα 3.12 που ακολουθεί.



1. First android device sends **sender id, application id** to GCM server for registration.
2. Upon successful registration GCM server issues **registration id** to android device.
3. After receiving registration id, device will send **registration id** to our server
4. Our server will store **registration id** in the **database** for later usage

- a Whenever push notification is needed, our server sends a **message** to GCM server along with device **registration id** (which is stored earlier in the database)
- b GCM server will delivers that message to respected mobile device using device registration id

Εικόνα 3.12. Γραφική αναπαράσταση λειτουργίας του GCM.

- ✚ ΒΗΜΑ 1: Στον server της google αρχικά στέλνεται το sender Id και το App Id από την Android συσκευή μας. Το sender Id είναι μοναδικό για κάθε συσκευή Android ενώ το App Id μας υποδεικνύει για πιο app θα γίνει η διαδικασία.
- ✚ ΒΗΜΑ 2: Αφού γίνει σωστά το registration στον GCM server, επιστρέφεται πίσω στην συσκευή ένα μοναδικό registration Id.
- ✚ ΒΗΜΑ 3: Αφού η συσκευή μας έχει το registration Id το στέλνει στον server μας.
- ✚ ΒΗΜΑ 4: Στη συνέχεια η συσκευή Android στέλνει αυτόν τον κωδικό στον server ο οποίος τον αποθηκεύουμε για μελλοντική χρήση.

Μία σημαντική προϋπόθεση για την επίτευξη των παραπάνω συμβάντων είναι να έχουμε google account.

3.4.4.3 Κρυπτογράφηση-Αποκρυπτογράφηση

Στην ανάπτυξη της συγκεκριμένης εφαρμογής με τίτλο «Προσφορά και ζήτηση αγοράς εργασίας» προκειμένου να φτάσουμε στο επίπεδο μιας επαγγελματικής εφαρμογής θα ήταν σκόπιμο να υπάρχει κρυπτογράφηση των ευαίσθητων δεδομένων όπως για παράδειγμα τον κωδικό πρόσβασης. Πιο συγκεκριμένα όταν αποστέλλεται ο κωδικός πρόσβασης στον server τον κρυπτογραφούμε και όταν φτάσει τον αποκρυπτογραφούμε με την γλώσσα προγραμματισμού PHP. Σύμφωνα με τα παραπάνω λοιπόν χρησιμοποιείται AES αλγόριθμος κρυπτογράφησης που είναι μέχρι στιγμής ο πολυπλοκότερος και πιο ασφαλής. Πιο αναλυτικά:

Κρυπτογραφία είναι ο επιστημονικός κλάδος που πραγματεύεται τη μελέτη και σχεδίαση κρυπτογραφικών τεχνικών, συστημάτων και πρωτοκόλλων. Μαζί με τον κλάδο της Κρυπτανάλυσης, που ασχολείται με τη μελέτη τρόπων παραβίασης αυτών, απαρτίζουν την Επιστήμη της Κρυπτολογίας. Έτσι, Κρυπτολογία είναι η επιστήμη της απόκρυψης, από τη μια πλευρά και, από την άλλη, της αποκάλυψης του περιεχομένου κωδικοποιημένων μηνυμάτων ή δεδομένων.

Η επιθυμία προστασίας του περιεχομένου μηνυμάτων μας οδήγησε στην επινοήση και χρήση κρυπτογραφικών τεχνικών και συστημάτων τα οποία

επιτρέπουν το μετασχηματισμό μηνυμάτων ή δεδομένων κατά τέτοιο τρόπο ώστε να είναι αδύνατη η υποκλοπή του περιεχομένου τους κατά τη μετάδοσή ή αποθήκευσή τους και, βεβαίως, την αντιστροφή του μετασχηματισμού. Η διαδικασία μετασχηματισμού καλείται κρυπτογράφηση και η αντιστροφή της αποκρυπτογράφηση.

Η συνάρτηση ή το σύνολο των κανόνων, στοιχείων και βημάτων που καθορίζουν την κρυπτογράφηση και την αποκρυπτογράφηση ονομάζεται κρυπτογραφικός αλγόριθμος. Η υλοποίηση του κρυπτογραφικού αλγόριθμου καλείται κρυπτογραφικό σύστημα. Μερικές φορές, ο κρυπτογραφικός αλγόριθμος καλείται και κωδικοποιητής (cipher). Πρωτόκολλα που χρησιμοποιούν κρυπτογραφικούς αλγόριθμους καλούνται κρυπτογραφικά πρωτόκολλα.

3.4.4.3.1 Στόχοι

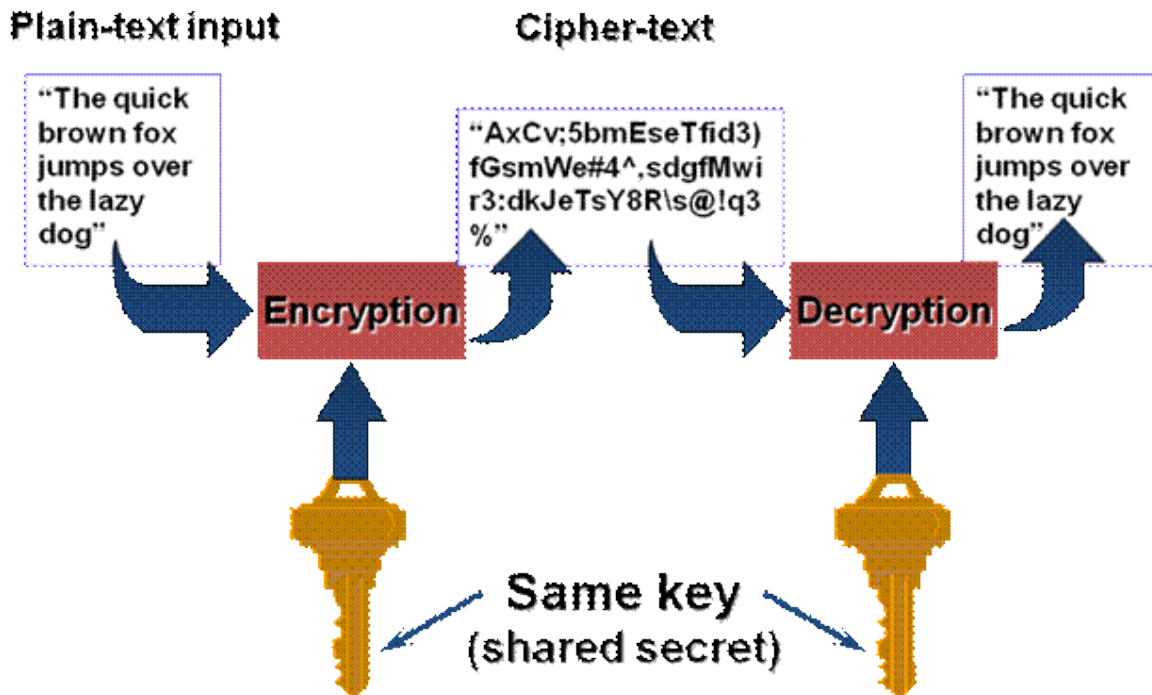
Η κρυπτογραφία παρέχει τέσσερις βασικές λειτουργίες («αντικειμενικοί σκοποί»):

- **Εμπιστευτικότητα:** Η πληροφορία προς μετάδοση είναι προσβάσιμη μόνο στα εξουσιοδοτημένα μέλη. Η πληροφορία είναι ακατανόητη σε κάποιον τρίτο.
- **Ακεραιότητα:** Η πληροφορία μπορεί να αλλοιωθεί μόνο από τα εξουσιοδοτημένα μέλη και δεν μπορεί να αλλοιώνεται χωρίς την ανίχνευση της αλλοίωσης.
- **Μη απάρνηση:** Ο αποστολέας ή ο παραλήπτης της πληροφορίας δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή της δημιουργίας της.
- **Πιστοποίηση:** Ο αποστολέας και ο παραλήπτης μπορούν να εξακριβώσουν τις ταυτότητές τους καθώς και την πηγή και τον προορισμό της πληροφορίας με διαβεβαίωση ότι οι ταυτότητές τους δεν είναι πλαστές.

3.4.4.3.2 Διαδικασία κρυπτογράφησης –αποκρυπτογράφησης

Η κρυπτογράφηση και αποκρυπτογράφηση ενός μηνύματος γίνεται με τη βοήθεια ενός αλγόριθμου κρυπτογράφησης (cipher) και ενός κλειδιού κρυπτογράφησης (key). Συνήθως ο αλγόριθμος κρυπτογράφησης είναι γνωστός,

οπότε η εμπιστευτικότητα του κρυπτογραφημένου μηνύματος που μεταδίδεται βασίζεται ως επί το πλείστον στη μυστικότητα του κλειδιού κρυπτογράφησης. Το μέγεθος του κλειδιού κρυπτογράφησης μετριέται σε αριθμό bits. Γενικά ισχύει ο εξής κανόνας: όσο μεγαλύτερο είναι το κλειδί κρυπτογράφησης, τόσο δυσκολότερα μπορεί να αποκρυπτογραφηθεί το κρυπτογραφημένο μήνυμα από επίδοξους εισβολείς. Διαφορετικοί αλγόριθμοι κρυπτογράφησης απαιτούν διαφορετικά μήκη κλειδιών για να πετύχουν το ίδιο επίπεδο ανθεκτικότητας κρυπτογράφησης.



Εικόνα 3.11 Διαδικασία κρυπτογράφηση- αποκρυπτογράφηση

3.5 Επίλογος

Συνοψίζοντας, σε αυτό το κεφάλαιο έγινε μια προσπάθεια να περιγραφτούν αναλυτικά οι γλώσσες προγραμματισμού, τα περιβάλλοντα ανάπτυξης αλλά και οι τεχνολογίες που χρησιμοποιήθηκαν προκειμένου να ολοκληρωθεί επιτυχώς το project. Σύμφωνα λοιπόν με όλα τα παραπάνω είμαστε έτοιμοι να περιγράψουμε την αναλυτική λειτουργία του.

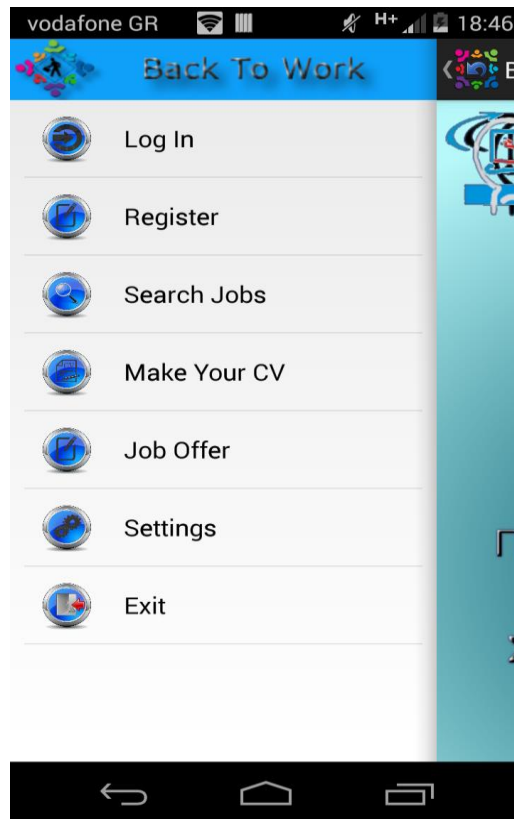
4 Αναλυτική λειτουργία και χρήση της εφαρμογής

4.1 Εισαγωγή

Στο τέταρτο και τελευταίο μέρος αυτής της πτυχιακής εργασίας θα γίνει εκτενή αναφορά στην αναλυτική λειτουργία της, με πολλά screen shot έτσι ώστε να γίνει κατανοητή από κάθε είδους αναγνώστη. Επιπρόσθετα σε αυτό το εισαγωγικό σημείο να επισημανθεί ότι όλα τα γραφικά συστατικά της εφαρμογής είναι σχεδιασμένα με το πρόγραμμα **Photoshop** και πέρα από αυτό όλα τα component και τα συμβάντα τους είναι σχεδιασμένα με .xml αρχεία για την βελτίωση του user expiring και για να μην θυμίζουν σε καμία περίπτωση τα απλά component που είναι προεγκατεστημένα στο Android.

4.2 Αρχική οθόνη

Στην εικόνα 4.1 παρουσιάζεται η αρχική οθόνη της εφαρμογής η οποία λαμβάνει χώρα αμέσως μετά την εκτέλεση της. Συγκεκριμένα βλέπουμε ένα List View στο οποίο παρουσιάζονται όλες οι λειτουργίες αυτής της εφαρμογής με την μορφή κουμπιών. Ακόμη όσο αφορά το προγραμματιστικό κομμάτι, παρατηρούμε 2 παράθυρα το ένα είναι όπως είπαμε το List View και το άλλο στα δεξιά είναι αυτό που ανοιγοκλείνει και αλλάζει κάθε φορά ανάλογα με το επιλεγόμενο menu δηλαδή το fragment μας. Όμως επειδή από αυτά που προαναφέρθηκαν, ένα fragment δεν μπορεί να στηθεί από μόνο του, πρέπει δηλαδή να έχει πάντα ένα Activity στο οποίο βασίζεται, και σκεπτόμενοι αυτό το αριστερό κομμάτι είναι το Activity. Αυτό πρακτικά σημαίνει ότι η εφαρμογή υλοποιήθηκε με καθαρά επαγγελματικά πρότυπα και άρα δεν δεσμεύεται η επεξεργαστική ισχύει του κινητού αλλά και της μνήμης ram με άσκοπο άνοιγμα Activity κάθε φορά, αλλά αντικατάσταση του ήδη υπάρχοντος. Το ανοιγόκλεισμα του fragment γίνεται με 2 τρόπους, πατώντας click στο Action bar είτε με σύρσιμο του παραθύρου(αριστερά-δεξιά)

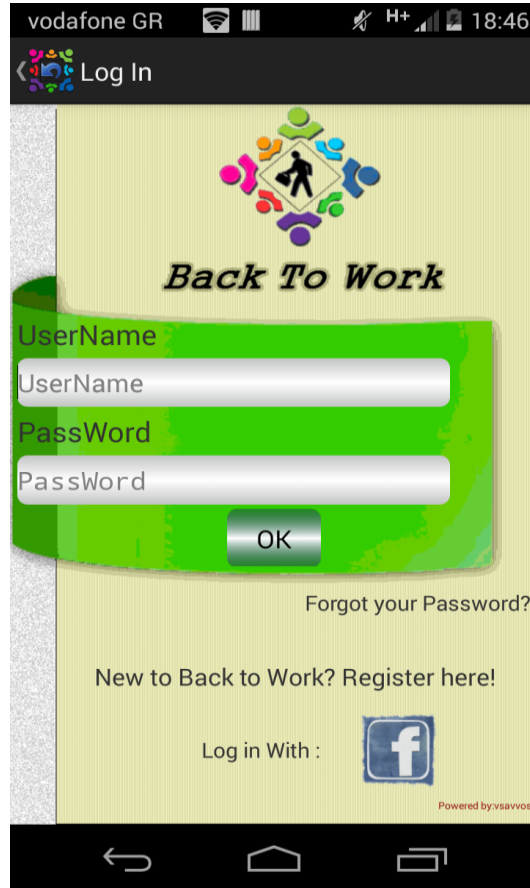


Εικόνα 4.1: Αρχική οθόνη εφαρμογής

Στην συνέχεια θα ήταν σκόπιμο να δούμε και να αναλύσουμε τα επιμέρους menu της εφαρμογής.

4.3 Log In Form

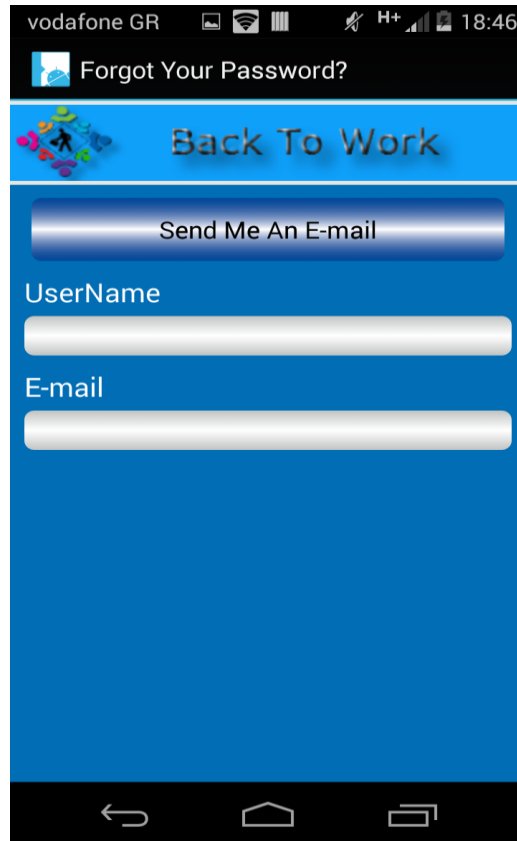
Σε αυτήν την φόρμα δίνεται η δυνατότητα στον χρήστη να κάνει log in με το user name και το password που διάλεξε ο ίδιος στην register form (θα αναλυθεί παρακάτω). Μόνο μετά την είσοδο του χρήστη στην εφαρμογή, μέσω αυτής της φόρμας, δίνεται η δυνατότητα να έχει πρόσβαση στις υπόλοιπες λειτουργίες της όπως στην δημιουργία βιογραφικού σημειώματος ή στην προσφορά εργασίας. Από την άλλη πλευρά η προσαρμοσμένη αναζήτηση θα είναι ανοιχτή σε όλους τους χρήστες είτε είναι μέλη είτε όχι αλλά μόνο για απλή προβολή αποτελεσμάτων χωρίς δηλαδή να υπάρχει επικοινωνία εργοδότη-εργαζομένου. Η οθόνη για το Log in προβάλλεται αμέσως μετά.



Εικόνα 4.2: Log In Form

4.3.1 Forgot your password Form

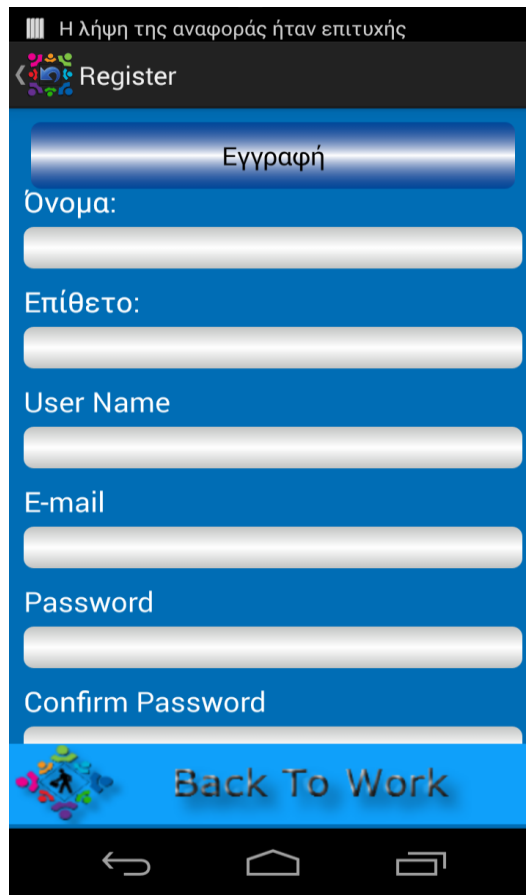
Ένα εύλογο ερώτημα είναι φυσικά τι γίνεται στην περίπτωση που ο χρήστης ξεχάσει τον κωδικό του με αποτέλεσμα να μην μπορεί να εισχωρήσει σε όλες τις λειτουργίες της εφαρμογής? Χάρης στην Forgot password form και με την συμπλήρωση των πεδίων Username και e-mail στέλνεται στον χρήστη e-mail με τον κωδικό του. Σίγουρα δεν παραλείπεται προγραμματιστικά ο έλεγχος «ταιριάσματος» του user name και του e-mail για την αποφυγή της εισαγωγής τυχαίων ηλεκτρονικών διευθύνσεων.



Εικόνα 4.2: Forgot password form

4.3.2 Register Form

Αυτή η φόρμα είναι η πρώτη φόρμα που θα πρέπει να συμπληρώσει ο χρήστης μετά την εγκατάσταση της εφαρμογής στην Android συσκευή του. Ουσιαστικά , είναι η φόρμα που δίνει την δυνατότητα να γίνει κάποιος μέλος στην εφαρμογή μετά από την συμπλήρωση των απαραίτητων πεδίων όπως όνομα, επώνυμο , password κ.α . Επιπρόσθετα σε όλες τις οθόνες υπάρχει το συστατικό στοιχείο Scroll View. Στιγμιότυπο της εικόνας (4.3) σας προβάλλουμε αμέσως μετά.

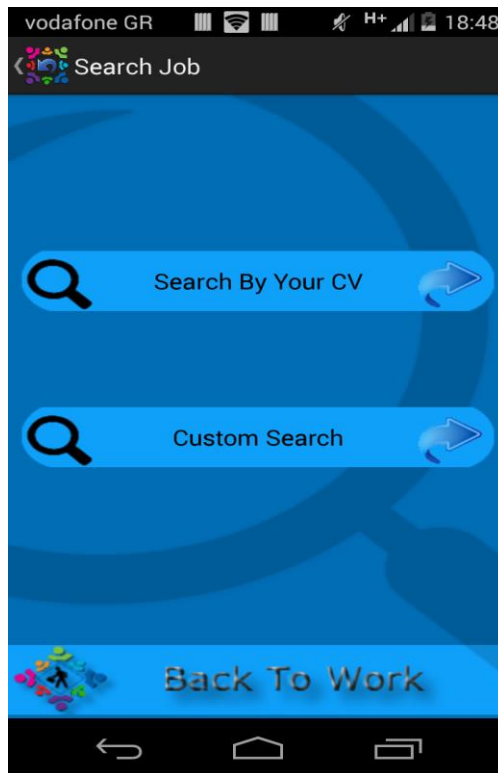


The image shows a mobile application interface for a registration form. At the top, there is a status bar with the text "Η λήψη της αναφοράς ήταν επιτυχής" and a hamburger menu icon. Below this is a dark header with a back arrow, a colorful icon, and the word "Register". The main form area has a blue background and contains several input fields: a blue button labeled "Εγγραφή", a text field for "Όνομα:", a text field for "Επίθετο:", a text field for "User Name", a text field for "E-mail", a text field for "Password", and a text field for "Confirm Password". At the bottom of the form is a blue button with a colorful icon and the text "Back To Work". The bottom of the screen shows a black navigation bar with three icons: a back arrow, a home icon, and a recent apps icon.

Εικόνα 4.3: Register Form

4.4 Search Form

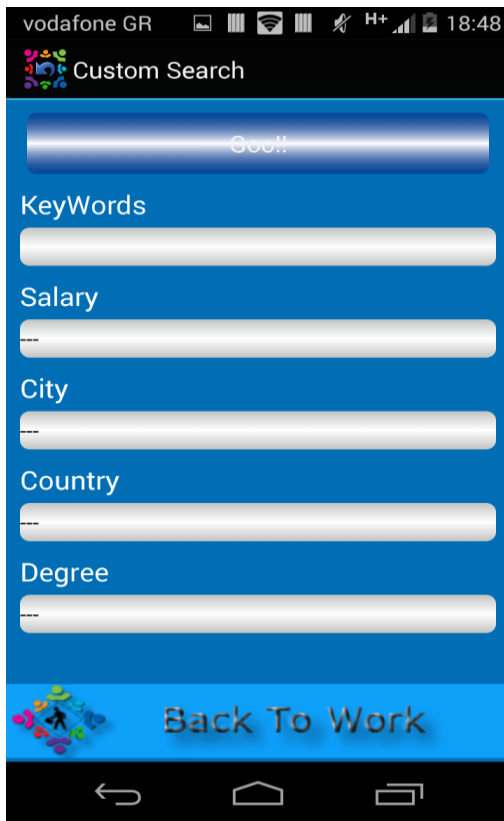
Ίσως αυτή η φόρμα είναι μία από τις σημαντικότερες φόρμες μιας και αυτό το σημείο είναι που ενδιαφέρει τον χρήστη. Αυτή η φόρμα δεν είναι άλλη από την αναζήτηση εργασίας. Πιο συγκεκριμένα , δίνεται στον χρήστη η επιλογή να επιλέξει τι αναζήτηση επιθυμεί ο ίδιος να κάνει. Οι επιλογές όπως φαίνεται και στην εικόνα 4.4 είναι δύο.



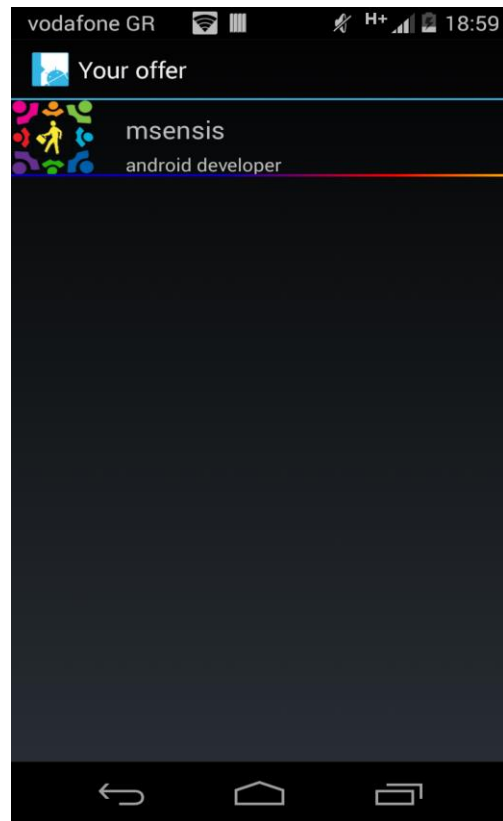
Εικόνα 4.4: Search Jobs Form

Η πρώτη επιλογή είναι μετά το πάτημα του κουμπιού να προβούμε στην διαδικασία εμφάνισης αποτελεσμάτων με τις προσφορές εργασίες αλλά μόνο εκείνων που ενδιαφέρουν τον χρήστη, για αυτό και προηγουμένως του ζητείται να συμπληρώσει το βιογραφικό του σημείωμα πράγμα που σημαίνει ότι εμφανίζονται στοχευμένα αποτελέσματα με γνώμονα το βιογραφικό του χρήστη.

Παρόλα αυτά αν κάποιος χρήστης δεν έχει πίεση χρόνου και θέλει να ρίξει μια πιο διεξοδική ματιά στην αγορά εργασίας του δίνεται η δυνατότητα να κάνει στοχευμένες αναζητήσεις που ο ίδιος επιθυμεί, με ένα ή περισσότερα πεδία της επιλογής του. Στιγμιότυπο της αναζήτησης αυτής εμφανίζεται παρακάτω.

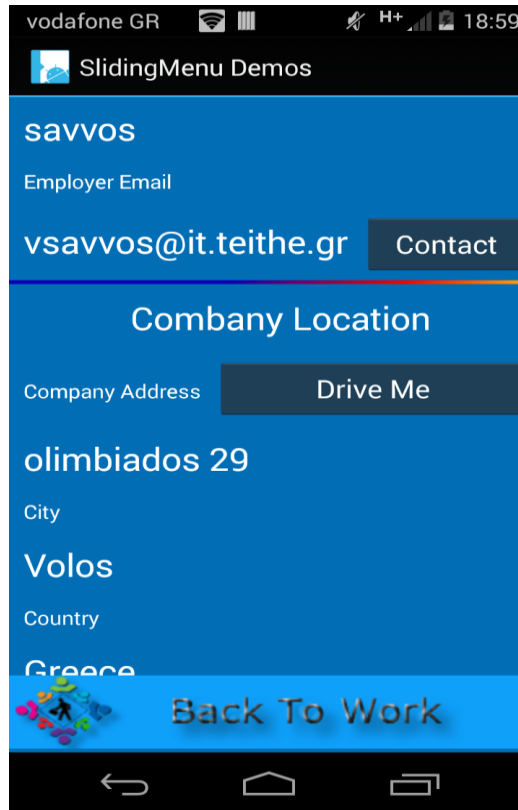


Εικόνα 4.5: Custom search



Εικόνα 4.6: Λίστα αποτελεσμάτων αναζήτησης

Μία λεπτομέρεια που αξίζει να σημειωθεί εδώ είναι το πεδίο key words. Αυτό πρακτικά σημαίνει ότι η εφαρμογή αναζητεί σε κάθε καταχωρημένη εργασία και σε όλα τα πεδία ή σχεδόν σε όλα την λέξη κλειδί που έχει πληκτρολογήσει ο χρήστης. Εν συνεχεία της διαδικασίας αναζήτησης αγοράς εργασίας εμφανίζουμε τα αποτελέσματα σε μορφή λίστας όπως για παράδειγμα στην εικόνα 4.6. Το αποτέλεσμα όλης αυτής της λειτουργίας είναι να έρθει σε επικοινωνία ο εργαζόμενος με τον εργοδότη και αυτό γίνεται με την φόρμα που ακολουθεί.

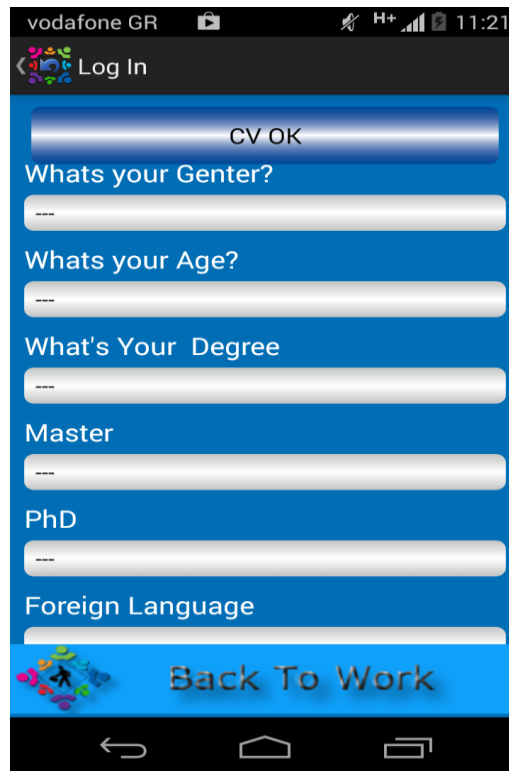


Εικόνα 4.7: Στοιχεία θέσης εργασίας

Στο στιγμιότυπο 4.7 προβάλλονται τα στοιχεία του εργοδότη, ο τόπος εργασίας, η χώρα, η αναλυτική περιγραφή των απαιτήσεων που έχει η εταιρία ως προς τους ενδιαφερόμενους και γενικά πολλές λεπτομέρειες για την προσφορά της θέσης εργασίας. Πέρα από αυτό όμως ο χρήστης έχει την δυνατότητα είτε να επικοινωνήσει με τον εργοδότη στέλνοντας e-mail ενδιαφέροντος με την επιλογή “contact” είτε να οδηγηθεί στον χώρο της εργασίας με την επιλογή “Drive Me” (για την επιλογή “Drive Me” χρησιμοποιώ το google map).

4.5 CV Form

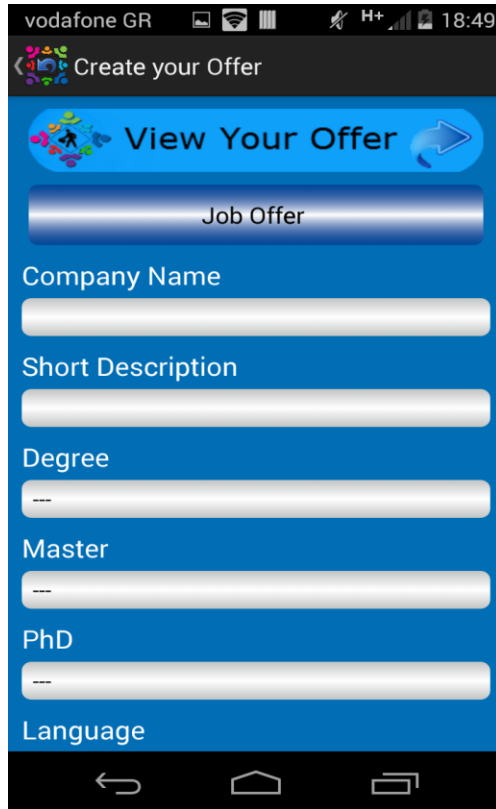
Σε αυτήν την επιλογή δίνεται η δυνατότητα στον χρήστη να δημιουργήσει το δικό του βιογραφικό από μια μεγάλη γκάμα επιλογών που αφορά την εκπαίδευση που έχει και διάφορα άλλα ενδιαφέροντα όπως το φίλο, το πτυχίο την μητρική γλώσσα κ.α.



Εικόνα 4.8: Φόρμα δημιουργίας βιογραφικού

4.6 Job Offer Form

Σε αυτή την τελευταία φόρμα, με τον ίδιο λογαριασμό ο χρήστης αν το επιθυμεί έχει την δυνατότητα να προσφέρει εργασία. Επιπλέον, από την στιγμή που δημιουργείται η νέα προσφορά εργασίας ταυτόχρονα ενημερώνονται και οι υποψήφιοι (ανάλογα με το βιογραφικό τους) εργαζόμενοι. Ακόμη ο χρήστης πιέζοντας το κουμπί «view your offer» είναι σε θέση να δει τις θέσεις προσφοράς εργασίας που έχει δημιουργήσει και να διαγράψει κάποια αν το επιθυμεί. Στιγμιότυπο της συγκεκριμένης φόρμας σας παρουσιάζω στην εικόνα (4.9) που ακολουθεί.

The image shows a mobile application interface for creating a job offer. At the top, the status bar displays 'vodafone GR', signal strength, Wi-Fi, and battery icons, along with the time '18:49'. Below the status bar is a navigation bar with a back arrow and the text 'Create your Offer'. The main content area has a blue background. It features a 'View Your Offer' button with a right-pointing arrow and a 'Job Offer' button. Below these are several input fields: 'Company Name', 'Short Description', 'Degree' (with a dropdown menu showing '---'), 'Master' (with a dropdown menu showing '---'), 'PhD' (with a dropdown menu showing '---'), and 'Language'. At the bottom, there is a black navigation bar with three icons: a back arrow, a home icon, and a recent apps icon.

Εικόνα 4.7: Φόρμα δημιουργίας θέσης εργασίας.

4.7 Επίλογος

Σε αυτό το κεφάλαιο, έγινε μια αναλυτική περιγραφή της λειτουργίας και χρήσης της εφαρμογής με screen shot και αναλυτική περιγραφή του καθενός. Στην συνέχεια θα αναφερθούν τα συμπεράσματα και οι μελλοντικές επεκτάσεις της εφαρμογής έτσι ώστε να γίνει πλήρως ανταγωνιστική και να συναγωνιστεί τις όμοιες της στο Google play store.

5 Επίλογος

5.1 Σύνοψη και συμπεράσματα

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μια εφαρμογή για κινητά τηλέφωνα τα οποία χρησιμοποιούν λειτουργικό Android, το οποίο όπως αναλύθηκε στα πρώτα κεφάλαια, είναι ένα ταχέως αναπτυσσόμενο λογισμικό το οποίο αυτή τη στιγμή κατέχει το μεγαλύτερο μέρος της αγοράς κινητών συσκευών. Επίσης, τα κινητά τηλέφωνα καθιερώνονται καθημερινά όλο και περισσότερο τόσο στην καταναλωτική αγορά όσο και στην καθημερινή χρήση δεκάδων διαφορετικών χρηστών. Αυτό καθιστά την επιλογή της ανάπτυξης της συγκεκριμένης εφαρμογής επιτυχή καθώς ανταποκρίνεται απόλυτα στις τρέχουσες τεχνολογικές εξελίξεις.

Η ενασχόληση με το πεδίο ανάπτυξης εφαρμογών για κινητές συσκευές και πιο συγκεκριμένα για συσκευές με λογισμικό Android κρίνεται άκρως ενδιαφέρουσα. Αποκομίσθηκαν πολύ σημαντικές γνώσεις τόσο για την σχεδίαση όσο και για τον προγραμματισμό εφαρμογών για κινητές συσκευές και έγιναν αντιληπτές οι διαφορές από τον παραδοσιακό προγραμματισμό για σταθερούς υπολογιστές.

Η ανάπτυξη εφαρμογών για κινητές συσκευές είναι ένα πεδίο το οποίο έχει την πολυτέλεια να βελτιώνεται, να αναπτύσσεται και να ανανεώνεται συνεχώς με την πάροδο του χρόνου. Αυτό, πέρα από την δημιουργία νέων θέσεων εργασίας, τόσο σε εταιρικό όσο και σε ανεξάρτητο προσωπικό επίπεδο, δημιουργεί και την ανάγκη για εύρεση νέων τεχνολογικών λύσεων στα νέα προβλήματα που θα προκύπτουν. Λύσεις που θα κληθούν να δώσουν οι νέοι προγραμματιστές και μηχανικοί.

5.2 Μελλοντικές επεκτάσεις

Παρά το ότι σε αυτό το στάδιο η εφαρμογή αναπτύχθηκε στα πλαίσια μιας πτυχιακής εργασίας και δεν είχε εμπορικό στόχο, θα μπορούσε εύκολα στο μέλλον μαζί με τις κατάλληλες τροποποιήσεις και επεκτάσεις να γίνει διαθέσιμη μέσω του

καταστήματος Play Store. Η εφαρμογή αναπτύχθηκε με τέτοιο τρόπο που καθιστά την επέκταση της πολύ εύκολη χωρίς να χρειαστούν αλλαγές στον υπάρχον κώδικα, παρά μόνο δημιουργία νέων κλάσεων και κώδικα που θα περιγράφει αυτές τις προσθήκες. Ακόμα, έχουν δημιουργηθεί ήδη οι βάσεις για συγκεκριμένες λειτουργίες που μπορούν εύκολα να προστεθούν στο μέλλον. Αυτές οι λειτουργίες είναι για παράδειγμα η εισαγωγή στην εφαρμογή με τα στοιχεία των social media όπως του facebook χωρίς να καθιστά απαραίτητη την registration form.

6 Βιβλιογραφία

Bill Phillips and Brian Hardy (2013), *Android Programming: The Big Nerd Ranch Guide*, Atlanta.

Donn Felker with Joshua Dobbs (2011), *Android Application Development for Dummies*, Canada.

Ed Burnette (2010), *Hello Android: Introducing Google's Mobile Development Platform*, North Carolina.

Jeff Friesen (2013), *Learn Java for Android Development, Second Edition*, New York.

Lauren Darcey and Shane Conder (2012), *Sams Teach Yourself Android Application Development in 24 Hours, Second Edition*, Indiana, 46240 USA.

Mark L. Murphy (2008), *The Busy Coder's Guide to Android Development*, United States of America.

Paul Deitel, Harvey Deitel and Abbey Deitel (2014), *Android for Programmers: An App-Driven Approach, 2nd Edition*, Canada.

Reto Meier (2012), *Professional Android 4 Application Development*, North Carolina.

Reto Meier (2010), *Android 2 Application Development*, United States of America.

Wei-Meng Lee (2012), *Beginning Android 4 Application Development*, Canada.

Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura (2011),
Programming Android: Java Programming for the New Generation of Mobile
Devices, United States of America.

Παράστημα A

Παρατίθεται ο κώδικας των εφαρμογών Android που αναπτύχθηκαν κατά την διάρκεια της εκπόνησης της πτυχιακής εργασίας.

Κρυπτογράφηση – αποκρυπτογράφηση στην java

```
public class AesEncDec {

    public static void main(String args[]) throws Exception{

        AesEncDec mdecrypt = new AesEncDec();
        /* Encrypt */
        String crypttext="1631989";
        String encrypted = AesEncDec.bytesToHex(
mdecrypt.encrypt(crypttext) );
        System.out.println("Encrypted text ::" + encrypted);

        /* Decrypt */
        String decrypted = new String( mdecrypt.decrypt(
"24a69580114b6910ce2910c471f5c204" ) ); // pairnei to parapano kai to vqazwei
kanoniko

        //String decrypted = new String(
mdecrypt.decrypt("8fe24ef3019aa1fc1c616b9de7f58f8c813c4bed986c1303beae6cf2b7337447
cdc5ce88ecfa854816dbdd81bdbb82d" ) );
        System.out.println("decrypted text ::" + decrypted);

    }

    private String iv = "fedcba9876543210";//Dummy iv (CHANGE IT!)
    private IvParameterSpec ivspec;
    private SecretKeySpec keyspec;
    private Cipher cipher;
    private String SecretKey = "0123456789abcdef";//Dummy secretKey
(CHANGE IT!)

    public AesEncDec()
    {
        ivspec = new IvParameterSpec(iv.getBytes());

        keyspec = new SecretKeySpec(SecretKey.getBytes(),
"AES");

        try {
            cipher = Cipher.getInstance("AES/CBC/NoPadding");
```

```

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

public byte[] encrypt(String text) throws Exception
{
    if(text == null || text.length() == 0)
        throw new Exception("Empty string");

    byte[] encrypted = null;

    try {
        cipher.init(Cipher.ENCRYPT_MODE, keyspec,
ivs spec);

        encrypted =
cipher.doFinal(padString(text).getBytes());
    } catch (Exception e)
    {
        throw new Exception("[encrypt] " +
e.getMessage());
    }

    return encrypted;
}

public byte[] decrypt(String code) throws Exception
{
    if(code == null || code.length() == 0)
        throw new Exception("Empty string");

    byte[] decrypted = null;

    try {
        cipher.init(Cipher.DECRYPT_MODE, keyspec,
ivs spec);

        decrypted = cipher.doFinal(hexToBytes(code));
        //Remove trailing zeroes
        if( decrypted.length > 0)
        {
            int trim = 0;
            for( int i = decrypted.length - 1; i >= 0;
i-- ) if( decrypted[i] == 0 ) trim++;

            if( trim > 0 )
            {
                byte[] newArray = new
byte[decrypted.length - trim];

```

```

        System.arraycopy(decrypted, 0, newArray,
0, decrypted.Length - trim);
        decrypted = newArray;
    }
}
} catch (Exception e)
{
    throw new Exception("[decrypt] " +
e.getMessage());
}
return decrypted;
}

```

```

public static String bytesToHex(byte[] data)
{
    if (data==null)
    {
        return null;
    }

    int len = data.Length;
    String str = "";
    for (int i=0; i<len; i++) {
        if ((data[i]&0xFF)<16)
            str = str + "0" +
java.Lang.Integer.toHexString(data[i]&0xFF);
        else
            str = str +
java.Lang.Integer.toHexString(data[i]&0xFF);
    }
    return str;
}

```

```

public static byte[] hexToBytes(String str) {
    if (str==null) {
        return null;
    } else if (str.Length() < 2) {
        return null;
    } else {
        int len = str.Length() / 2;
        byte[] buffer = new byte[len];
        for (int i=0; i<len; i++) {
            buffer[i] = (byte)
Integer.parseInt(str.Substring(i*2,i*2+2),16);
//System.out.println("τι είναι αυτο "+
Integer.parseInt(str.Substring(i*2,i*2+2),16));
        }
        return buffer;
    }
}
}

```

```
private static String padString(String source)
{
    char paddingChar = ' ';
    int size = 16;
    int x = source.length() % size;
    int padLength = size - x;

    for (int i = 0; i < padLength; i++)
    {
        source += paddingChar;
    }

    return source;
}
}
```

κρυπτογράφηση – αποκρυπτογράφηση στην PHP

```
class MCrypt
{
    private $iv = 'fedcba9876543210'; #Same as in JAVA
    private $key = '0123456789abcdef'; #Same as in JAVA

    function __construct()
    {
    }

    function encrypt($str) {

        //$key = $this->hex2bin($key);
        $iv = $this->iv;

        $td = mcrypt_module_open('rijndael-128', '', 'cbc', $iv);

        mcrypt_generic_init($td, $this->key, $iv);
        $encrypted = mcrypt_generic($td, $str);
    }
}
```

```
mdecrypt_generic_deinit($td);
mdecrypt_module_close($td);

return bin2hex($encrypted);
}

function decrypt($code) {
//$key = $this->hex2bin($key);
$code = $this->hex2bin($code);
$iv = $this->iv;

$td = mdecrypt_module_open('rijndael-128', '', 'cbc', $iv);

mdecrypt_generic_init($td, $this->key, $iv);
$decrypted = mdecrypt_generic($td, $code);

mdecrypt_generic_deinit($td);
mdecrypt_module_close($td);

return utf8_encode(trim($decrypted));
}

protected function hex2bin($hexdata) {
$bindata = '';

for ($i = 0; $i < strlen($hexdata); $i += 2) {
$bindata .= chr(hexdec(substr($hexdata, $i, 2)));
}

return $bindata;
}
```

E-mail validate

```
public class EmailFormatValidator {

    private Pattern pattern;
    private Matcher matcher;

    private static final String EMAIL_PATTERN = "[_A-Za-z0-9-\\+]+(\\.[_A-
Za-z0-9-]+)*@"
        + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";

    public EmailFormatValidator() {
        pattern = Pattern.compile(EMAIL_PATTERN);
    }

    public boolean validate(final String email) {

        matcher = pattern.matcher(email);
        return matcher.matches();

    }
}
```

Fragment transaction

```
switch (position) {

    case 0:

        log_in_Fragment fragment = new
        log_in_Fragment();
        fragmentManager =
        getFragmentManager();
        fragmentTransaction =
        fragmentManager.beginTransaction();
        fragmentTransaction.replace(R.id.mesasto, fragment); // mesasto vale to
        kenourio fragment
        //fragmentTransaction.addToBackStack(null);
        fragmentTransaction.commit();

        break;

    case 1:

        register_Fragment registerFragclas
        = new register_Fragment();
        fragmentManager =
        getFragmentManager();
```

```

fragmentTransaction =
fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.mesasto, registerFragclas); // mesasto vale
to kenourio fragment
//
fragmentTransaction.addToBackStack(null);
fragmentTransaction.commit();

break;
case 2:

search_Fragment searchFragclas = new
search_Fragment();
fragmentManager =
getFragmentManager();
fragmentTransaction =
fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.mesasto, searchFragclas);
//
fragmentTransaction.addToBackStack(null);
fragmentTransaction.commit();
break;
case 10:
// favourite Fragment

break;
case 3:

cv_Fragment cvFragclas = new
cv_Fragment();
fragmentManager =
getFragmentManager();
fragmentTransaction =
fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.mesasto, cvFragclas); // mesasto vale to
kenourio fragment
//fragmentTransaction.addToBackStack(null);
fragmentTransaction.commit();
break;
case 4:

// myActivity = (CustomAnimation)
(getActivity());
//
(myActivity.findViewById(R.id.imageviewmesasto)).setVisibility(getView().
GONE); // gia na figei to image view kai na min kolaei
//
//job offer

```

```

        Fragment joboferFragclas = new
joboffer_Fragment();
        fragmentManager =
getFragmentManager();
        fragmentManager.beginTransaction();
fragmentTransaction.replace(R.id.mesasto, joboferFragclas); // mesasto vale
to kenourio fragment
//fragmentTransaction.addToBackStack(null);
        fragmentManager.commit();

        break;
    case 5:
        //settings

        Intent intent = new
Intent(getActivity(),RegisterActivity.class);
        startActivity(intent);
        Log.e("gia poio logo anoigei re",
"gia poio logo anoigei re");

        break;
    case 6:

        getActivity().finish();

        getActivity().overridePendingTransition(R.anim.slide_in_bottom,
R.anim.slide_out_top);

        break;
    }

```

Alert Dialog

```

public class AlertDialogManager {
    /**
     * Function to display simple Alert Dialog
     * @param context - application context
     * @param title - alert dialog title
     * @param message - alert message
     * @param status - success/failure (used to set icon)
     * - pass null if you don't want icon
     */
    public void showAlertDialog(Context context, String title, String
message,
        Boolean status) {
        AlertDialog alertDialog = new
AlertDialog.Builder(context).create();

        // Setting Dialog Title
        alertDialog.setTitle(title);

```



```
// Setting Dialog Message
alertDialog.setMessage(message);

if(status != null)
    // Setting alert dialog icon
    alertDialog.setIcon((status) ? R.drawable.success :
R.drawable.fail);

// Setting OK Button
alertDialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
    }
});

// Showing Alert Message
alertDialog.show();
}
}
```

Base Activity

```
public class BaseActivity extends SlidingFragmentActivity {

    private int mTitleRes;
    protected ListFragment mFrag;

    public BaseActivity(int titleRes) {
        mTitleRes = titleRes;
    }

    public BaseActivity(){}

    public void onBackPressed()
    {
        if (!getSlidingMenu().isShown()) {
            super.toggle();
        } else {
            super.onBackPressed(); // invoke super's method or other logic?
        }
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //setTitle(mTitleRes);
    }
}
```

```

setTitle("Base Menu");

// set the Behind View
setBehindContentView(R.layout.menu_frame);

getSupportActionBar().setLogo(R.drawable.Logoback);

if (savedInstanceState == null) {
    FragmentTransaction t =
this.getSupportFragmentManager().beginTransaction();

ListFragment mFrag = new SampleListFragment(); // paizei bala kai i
t.replace(R.id.menu_frame, mFrag);
t.commit();
} else {
    mFrag =
(ListFragment)this.getSupportFragmentManager().findFragmentById(R.id.menu_frame)
;
}

// customize the SlidingMenu
// auta einai na min feugei olo to fragment aristera kai dexia
SlidingMenu sm = getSlidingMenu();
sm.setShadowWidthRes(R.dimen.shadow_width);
sm.setShadowDrawable(R.drawable.shadow);
sm.setBehindOffsetRes(R.dimen.slidingmenu_offset);
sm.setFadeDegree(0.35f);
sm.setTouchModeAbove(SlidingMenu.TOUCHMODE_FULLSCREEN);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

Log.e(" mesaaa sto Base activity", " mesaaa sto Base activity");
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            // dose ena poli mikro delay

            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    toggle();
                }
            }, 25);

            return true;
        case R.id.github:
    }
}

```

```
        Util.goToGitHub(this);
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

Connection Detector

```
public class ConnectionDetector {

    private Context _context;

    public ConnectionDetector(Context context){
        this._context = context;
    }

    /**
     * Checking for all possible internet providers
     * **/
    public boolean isConnectingToInternet(){
        ConnectivityManager connectivity = (ConnectivityManager)
        _context.getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivity != null)
        {
            NetworkInfo[] info = connectivity.getAllNetworkInfo();
            if (info != null)
                for (int i = 0; i < info.length; i++)
                    if (info[i].getState() == NetworkInfo.State.CONNECTED)
                    {
                        return true;
                    }
        }
        return false;
    }
}
```

Register Fragment

```
public class register_Fragment extends Fragment { //implements
interface RePaidiaKanountinzoieukoli {

    View view;
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        getActivity().setTitle("Register");
        view = inflater.inflate(R.layout.register_layout ,container,
false);

        return view;
    }
}
```

```

//METAVLITES METAVLITES METAVLITES METAVLITES METAVLITES METAVLITES
METAVLITES METAVLITES METAVLITES
    int existssuccess=11;
    Button btnregister;
    ProgressDialog progressDialog;
    Builder alert;
    EditText
editusnamereg,editonomaregister,editepithetoregister,editPasswordregister;
    EditText editConfirmPassregister;
    EditText editemailregister;
    Boolean olakala=false;

    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);

        btnregister= (Button)
view.findViewById(R.id.btbregisterButton);
        btnregister.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                //oles oi metavlites p xrisimopoiounte na
                pernoun ta default
                existssuccess=11;
                olakala=false;

editusnamereg=(EditText)view.findViewById(R.id.editUserNameregister);
editonomaregister=(EditText)view.findViewById(R.id.editonomaregister);
editepithetoregister=(EditText)view.findViewById(R.id.editepithetoregister);
editPasswordregister=(EditText)view.findViewById(R.id.editPasswordregister);
editConfirmPassregister=(EditText)view.findViewById(R.id.editConfirmPassregister
);
editemailregister=(EditText)view.findViewById(R.id.editemailregister);

                Log.e("ti timiiii exei re gamotoooo",
                existssuccess+ " malista");

                progressDialog =
ProgressDialog.show(getActivity() , "Register..", "Loading...");
                alert = new
AlertDialog.Builder(getActivity());
                alert.setIcon(R.drawable.atension);

                new Thread() {

```

```
public void run() {  
    try {  
        jsonregister(); //  
        mesa se ena thread  
        Log.e("to return to  
        stelnei edo ", "to return to stelnei edo");  
    } catch (Exception e) {  
        Log.e("giati  
        exceptionnnn", e.getMessage()+" ");  
        Log.e("poteeeeeeeeeeeeeeeeeee ", "edoooooooooooooooooooooooooooo");  
        progressDialog.dismiss();  
        getActivity().runOnUiThread(new Runnable() { // oreo to thredaki allages sto  
        gui  
        @Override  
        public  
        void run() {  
            alert.setTitle("ERROR");  
            alert.setMessage("Check your connection");  
            alert.setPositiveButton("OK", null);  
            alert.show();  
        }  
    });  
    }  
    if(existssuccess==1){  
        progressDialog.dismiss();  
        getActivity().runOnUiThread(new Runnable() { // oreo to thredaki allages sto  
        gui  
        @Override  
        public  
        void run() {  
            alert.setTitle("Hmmmmmmmm");
```

```
alert.setMessage("Iparxei idi tetoia egrafi stin vasi");

alert.setPositiveButton("OK", null);

alert.show();

});
}

if(olakala==true){

progressDialog.dismiss();

getActivity().runOnUiThread(new Runnable() { // oreo to thredaki allages sto
gui

@Override

public

void run() {

alert.setTitle("Congrantulation!!");

alert.setMessage("Moli eginas mellos");

alert.setPositiveButton("OK", null);

alert.show();

});
}

else{

progressDialog.dismiss();}

}

}.start();

}

});
// telos tou button click
```

```

    }
    //(editPasswordregister.getText().length()!=0 &&
    editConfirmPassregister.getText().length()!=0 )||

        //(! (editPasswordregister.getText().toString().equals(editConfirmPassregi
        ster.getText().toString()))
        public void jsonregister() throws JSONException{
            Log.e("megethosssssssssss",
            editPasswordregister.getText().toString().length() + "");
            if(editPasswordregister.getText().length()==0
            ||(! (editPasswordregister.getText().toString().equals(editConfirmPassregister.ge
            tText().toString()))){
                getActivity().runOnUiThread(new Runnable() { // oreo to
                thredaki allages sto gui

                    @Override
                    public void run() {
                        alert.setTitle("ERROR");
                        alert.setMessage("The tow passwords are
                        not equals");

                        alert.setPositiveButton("OK", null);
                        alert.show();

                    }
                });
                return;

            }

            if(editusernameg.getText().toString().length()==0){
                getActivity().runOnUiThread(new Runnable() { // oreo
                to thredaki allages sto gui

                    @Override
                    public void run() {
                        alert.setTitle("ERROR");
                        alert.setMessage("Check your username");
                        alert.setPositiveButton("OK", null);
                        alert.show();

                    }
                });
                return;

            }

            if(editonomaregister.getText().toString().length()==0){
                getActivity().runOnUiThread(new Runnable() { // oreo
                to thredaki allages sto gui

                    @Override
                    public void run() {
                        alert.setTitle("ERROR");
                        alert.setMessage("Check your name");
                    }
                });
            }
        }
    }

```

```
        alert.setPositiveButton("OK", null);
        alert.show();
    }
});
return;
}

if(editepithetoregister.getText().toString().length()==0){
    getActivity().runOnUiThread(new Runnable() { // oreo
to thredaki allages sto gui

        @Override
        public void run() {
            alert.setTitle("ERROR");
            alert.setMessage("Check your surname");
            alert.setPositiveButton("OK", null);
            alert.show();
        }
    });
return;
}

if(editemailregister.getText().toString().length()==0){
    getActivity().runOnUiThread(new Runnable() { // oreo
to thredaki allages sto gui

        @Override
        public void run() {
            alert.setTitle("ERROR");
            alert.setMessage("Check your email");
            alert.setPositiveButton("OK", null);
            alert.show();
        }
    });
return;
}

if(editPasswordregister.getText().toString().length()==0){
    getActivity().runOnUiThread(new Runnable() { // oreo
to thredaki allages sto gui

        @Override
        public void run() {
            alert.setTitle("ERROR");
            alert.setMessage("Check your password");
            alert.setPositiveButton("OK", null);
            alert.show();
        }
    });
return;
}
```



```
Log.e("auto den prepei nato emfanisei", "auto den prepei na to
emfanisei");

//usernameonadiko
JSONParser httpRequest = new JSONParser();
List<NameValuePair> params3 = new
ArrayList<NameValuePair>();//mitsos
params3.add(new BasicNameValuePair("username1",
editusernameregister.getText().toString());//mitsos
JSONObject jsonFromServer1
=httpRequest.makeHttpRequest("http://testdev.freeiz.com/checkifecists.php",
"POST", params3);//mitsos

existssuccess = jsonFromServer1.getInt("success"); //i miden
i ena
Log.e("to sucess einai den iparxei tetio username: "+
existssuccess, "to sucess einai den iparxei tetio username" + existssuccess);

if(existssuccess==0){

JSONParser httpRequest1 = new JSONParser();
List<NameValuePair> params4= new
ArrayList<NameValuePair>();//mitsos
params4.add(new
BasicNameValuePair("onoma", editonomaregister.getText().toString());//mitsos
params4.add(new
BasicNameValuePair("eponimo",
editepithetoregister.getText().toString());//mitsos
params4.add(new
BasicNameValuePair("email", editemailregister.getText().toString());//mitsos
params4.add(new
BasicNameValuePair("username", editusernameregister.getText().toString());//mitsos
params4.add(new
BasicNameValuePair("password",
editPasswordregister.getText().toString());//mitsos
//elengxei me java an ola simplirothikan

httpRequest.makeHttpRequest("http://testdev.freeiz.com/register1.php",
"POST", params4);

olakala=true;

};//telos if
}
```

Http Post Request με Java

```
JSONParser httpRequest = new JSONParser();
List<NameValuePair> params3 = new
ArrayList<NameValuePair>();//mitsos
params3.add(new BasicNameValuePair("username1",
editusername1.getText().toString()));//mitsos
JSONObject jsonFromServer1
=httpRequest.makeHttpRequest("http://testdev.freeiz.com/checkifecists.php",
"POST", params3);//mitsos

        existssuccess = jsonFromServer1.getInt("success"); //i miden
i ena
        Log.e("to sucess einai den iparxei tetio username: "+
existssuccess, "to sucess einai den iparxei tetio username" + existssuccess);

        if(existssuccess==0){

                JSONParser httpRequest1 = new JSONParser();
                List<NameValuePair> params4= new
ArrayList<NameValuePair>();//mitsos
                params4.add(new
BasicNameValuePair("onoma", editonomaregister.getText().toString()));//mitsos
                params4.add(new
BasicNameValuePair("eponimo",
editepithetoregister.getText().toString());//mitsos
                params4.add(new
BasicNameValuePair("email", editemailregister.getText().toString());//mitsos
                params4.add(new
BasicNameValuePair("username", editusername1.getText().toString());//mitsos
                params4.add(new
BasicNameValuePair("password",
editpasswordregister.getText().toString());//mitsos
                //elengxei me java an ola simplirothikan

                httpRequest.makeHttpRequest("http://testdev.freeiz.com/register1.php",
"POST", params4);

                olakala=true;
```

GCM Registration

```
public class MainActivity extends Activity {
    // label to display gcm messages
    //TextView lblMessage;
```

```

// AsyncTask
AsyncTask<Void, Void, Void> mRegisterTask;

// Alert dialog manager
AlertDialogManager alert = new AlertDialogManager();

// Connection detector
ConnectionDetector cd;

//public static String name;
//public static String email;

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    cd = new ConnectionDetector(getApplicationContext());

    // Check if Internet present
    if (!cd.isConnectingToInternet()) {
        // Internet Connection is not present
        alert.showAlertDialog(MainActivity.this,
            "Internet Connection Error",
            "Please connect to working Internet
connection", false);
        // stop executing code by return
        return;
    }

    // Getting name, email from intent
    //Intent i = getIntent();

    //name = i.getStringExtra("name");
    //email = i.getStringExtra("email");

    // Make sure the device has the proper dependencies.
    GCMRegistrar.checkDevice(this);

    // Make sure the manifest was properly set - comment out this line
    // while developing the app, then uncomment it when it's ready.
    GCMRegistrar.checkManifest(this);

    //lblMessage = (TextView) findViewById(R.id.lblMessage);

    registerReceiver(mHandleMessageReceiver, new IntentFilter(
        DISPLAY_MESSAGE_ACTION));

    // Get GCM registration id
    final String regId = GCMRegistrar.getRegistrationId(this);

    // Check if regid already presents
    if (regId.equals("")) {
        // Registration is not present, register now with GCM

```

```

        GCMRegistrar.register(this, SENDER_ID);
    } else {
        // Device is already registered on GCM
        if (GCMRegistrar.isRegisteredOnServer(this)) {
            // Skips registration.
            Toast.makeText(getApplicationContext(), "Already
registered with GCM", Toast.LENGTH_LONG).show();
        } else {
            // Try to register again, but not in the UI thread.
            // It's also necessary to cancel the thread

onDestroy(),

            // hence the use of AsyncTask instead of a raw thread.
            final Context context = this;
            mRegisterTask = new AsyncTask<Void, Void, Void>() {

                @Override
                protected Void doInBackground(Void... params) {
                    // Register on our server
                    // On server creates a new user
                    ServerUtilities.register(context,
"nothing", "nuthing", regId); //server utilities class kanei to register
                    return null;
                }

                @Override
                protected void onPostExecute(Void result) {
                    mRegisterTask = null;
                }

            };
            mRegisterTask.execute(null, null, null);
        }
    }
}

/**
 * Receiving push messages
 */
private final BroadcastReceiver mHandleMessageReceiver = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String newMessage =
intent.getExtras().getString(EXTRA_MESSAGE);
        // Waking up mobile if it is sleeping
        WakeLocker.acquire(getApplicationContext());

        /**
         * Take appropriate action on this message
         * depending upon your app requirement
         * For now i am just displaying it on the screen
         */

        // Showing received message
        //lblMessage.append(newMessage + "\n");
    }
}

```

```

        Toast.makeText(getApplicationContext(), "New Message: " +
newMessage, Toast.LENGTH_LONG).show();

        // Releasing wake lock
        WakeLocker.release();
    }
};

@Override
protected void onDestroy() {
    if (mRegisterTask != null) {
        mRegisterTask.cancel(true);
    }
    try {
        unregisterReceiver(mHandleMessageReceiver);
        GCMRegistrar.onDestroy(this);
    } catch (Exception e) {
        Log.e("UnRegister Receiver Error", "> " + e.getMessage());
    }
    super.onDestroy();
}
}
}

```

Json Parser

```

@TargetApi(Build.VERSION_CODES.GINGERBREAD)
@SuppressLint("NewApi")
public class JSONParser {

    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    @TargetApi(Build.VERSION_CODES.GINGERBREAD)
    // by making HTTP POST or GET method
    @SuppressLint("NewApi")
    public JSONObject makeHttpRequest(String url, String
method, List<NameValuePair> params) {
        if (android.os.Build.VERSION.SDK_INT > 9) {
            StrictMode.ThreadPolicy policy =
                new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
        }
        // Making HTTP request
        try {

```

```
// check for request method
if(method == "POST"){
    // request method is POST
    // defaultHttpClient
    DefaultHttpClient httpClient = new
DefaultHttpClient();

    HttpPost httpPost = new HttpPost(url);

    httpPost.setEntity(new UrlEncodedFormEntity(params));

    HttpResponse httpResponse =
httpClient.execute(httpPost);
    HttpEntity httpEntity = httpResponse.getEntity();
    is = httpEntity.getContent();

}else if(method == "GET"){
    // request method is GET

    DefaultHttpClient httpClient = new
DefaultHttpClient();

    String paramString = URLEncodedUtils.format(params,
"utf-8");

    url += "?" + paramString;

   HttpGet httpGet = new HttpGet(url);

    HttpResponse httpResponse =
httpClient.execute(httpGet);

    HttpEntity httpEntity = httpResponse.getEntity();

    is = httpEntity.getContent();

}

} catch (UnsupportedEncodingException e) {

    Log.e("-3", e.toString() );
} catch (ClientProtocolException e) {
    Log.e("-2", e.toString() );
} catch (IOException e) {
    Log.e("-1", e.toString() );
}
}
catch(Exception e)
{
    Log.e("-4", e.toString() );
}
}
```

```
try {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        is, "UTF-8"), 8);
    //iso-8859-7
    StringBuilder sb = new StringBuilder();

    String line = null;

    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    Log.e("", "= " +sb.toString());
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " +
e.toString());
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " + e.toString());
}
{
    catch(Exception e)
}

}

// return JSON String
return jsonObj;
}
```

CustomSearch.php

<?php

```
require_once('DBcon.php');
$response = array();
$variable1 = $_POST['keywords'];
$variable2 = $_POST['salary'];
$variable3 = $_POST['city'];
$variable4 = $_POST['degree'];

//$variable1 = 'android';
//$variable2 = '---';
```

```
//$variable3 = '---';  
//$variable4 = '---';  
  
// "" (an empty string)  
// 0 (0 as an integer)  
// 0.0 (0 as a float)  
// "0" (0 as a string)  
// NULL  
// FALSE  
// array() (an empty array)  
// $var; (a variable declared, but without a value)  
  
if($variable1==""){  
    $var1='0';    // AAAsimplioto  
}  
else{  
    $var1='1';    // simpliromeno  
}  
  
if($variable2=='---'){  
    $var2='0';  
}  
else{  
    $var2='1';  
}  
  
if($variable3=='---'){  
    $var3='0';  
}  
else{  
    $var3='1';  
}  
  
if($variable4=='---'){ //degree idio  
    $var4='0';  
}  
else{  
    $var4='1';  
}
```



```

$variable5=$var1.$var2.$var3.$var4;

//echo $variable5."---";
// echo $variable1."---";

if ($variable5 == '0001') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDegree='$variable4' ") or die(mysql_error());
    //echo "1";
}
elseif ($variable5 == '0010') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferCity='$variable3' ") or die(mysql_error());
    //echo "2";
}
elseif ($variable5 == '0011') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferCity='$variable3' and
jobOfferDegree='$variable4' ") or die(mysql_error());

    //echo "3";
}
elseif ($variable5 == '0100') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferSalary='$variable2' ") or die(mysql_error());

    // echo "4";
}
elseif ($variable5 == '0101') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferSalary='$variable2' and
jobOfferDegree='$variable4' ") or die(mysql_error());

    // echo "5";
}
elseif ($variable5 == '0110') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =

```

```

jobOffer.jobOfferId and jobOfferSalary='$variable2' and
jobOfferCity='$variable3' ") or die(mysql_error());

    // echo "6";
}
elseif ($variable5 == '0111') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferSalary='$variable2' and
jobOfferCity='$variable3' and jobOfferDegree='$variable4' ") or
die(mysql_error());

    // echo "7";
}
elseif ($variable5 == '1000') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' ") or
die(mysql_error());

    // echo "8";
}
elseif ($variable5 == '1001') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' and
jobOfferDegree='$variable4' ") or die(mysql_error());
    // echo "9";
}
elseif ($variable5 == '1010') {
    $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' and
jobOfferCity='$variable3' ") or die(mysql_error());

    // echo "10";
}
elseif ($variable5 == '1011') {
    $result = mysql_query("SELECT * FROM ergodotis ,users ,
jobOffer where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' and
jobOfferCity='$variable3' and jobOfferDegree='$variable4' ") or die(mysql_error());

    // echo "11";
}
elseif ($variable5 == '1100') {

```

```

        $result = mysql_query("SELECT * FROM ergodotis ,users , jobOffer
where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' and
jobOfferSalary='$variable2' ") or die(mysql_error());

        // echo "12";
    }
    elseif ($variable5 == '1101') {
        $result = mysql_query("SELECT * FROM ergodotis ,users ,
jobOffer where users.userid=ergodotis.IdRefuser and ergodotis.useridergodoti =
jobOffer.jobOfferId and jobOfferDe LIKE '%$variable1%' and
jobOfferSalary='$variable2' and jobOfferDegree='$variable4' ") or
die(mysql_error());

        // echo "13";
    }
    elseif ($variable5 == '1110') {
        $result = mysql_query("SELECT * FROM ergodotis
,users , jobOffer where users.userid=ergodotis.IdRefuser and
ergodotis.useridergodoti = jobOffer.jobOfferId and jobOfferDe LIKE
'%$variable1%' and jobOfferSalary='$variable2' and jobOfferCity='$variable3' ") or
die(mysql_error());

        // echo "14";
    }
    elseif ($variable5 == '1111') {
        $result = mysql_query("SELECT * FROM
ergodotis ,users , jobOffer where users.userid=ergodotis.IdRefuser and
ergodotis.useridergodoti = jobOffer.jobOfferId and jobOfferDe LIKE
'%$variable1%' and jobOfferSalary='$variable2' and jobOfferCity='$variable3' and
jobOfferDegree='$variable4' ") or die(mysql_error());

        // echo "15";
    }

    if (mysql_num_rows($result) !=0){

        $response["user"] = array();

        while ($row = mysql_fetch_array($result)) {
            // temp user array
            $product = array();

```

```
        //onomaEtairias ,  sxolioErgdoti ,onoma ,
eponimo, email , odos , jobOfferSalary, jobOfferCity,
jobOfferCountry,jobOfferDegree,
        //jobOfferMaster,  jobOfferPhd ,jobOfferLanguage ,
jobOfferArmy ,  jobOfferDe

    $product["onomaEtairias"] = $row["onomaEtairias"];
    $product["sxolioErgdoti"] = $row["sxolioErgdoti"];
    $product["onoma"] = $row["onoma"];
    $product["eponimo"] = $row["eponimo"];
    $product["email"] = $row["email"];
    $product["odos"] = $row["odos"];
    $product["jobOfferSalary"] = $row["jobOfferSalary"];
    $product["jobOfferCity"] = $row["jobOfferCity"];
    $product["jobOfferCountry"] = $row["jobOfferCountry"];
    $product["jobOfferDegree"] = $row["jobOfferDegree"];
    $product["jobOfferMaster"] = $row["jobOfferMaster"];
    $product["jobOfferPhd"] = $row["jobOfferPhd"];
    $product["jobOfferLanguage"] = $row["jobOfferLanguage"];
    $product["jobOfferArmy"] = $row["jobOfferArmy"];
    $product["jobOfferDe"] = $row["jobOfferDe"];

    // push single product into final response array
    array_push($response["user"], $product);
}
// success
$response["success"] = 1;

// echoing JSON response
echo json_encode($response);

} else {
    // no products found
    $response["success"] = 0;

    // echo no users JSON
    echo json_encode($response);
}
```