



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Android Εφαρμογής «Δίαιτα (DietDroid)» Development of Android Application «Diet (DietDroid)»
Όνοματεπώνυμο Φοιτητή	Αντώνιος Μανέππας
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	ΜΠΠΛ/ 10004
Επιβλέπων	Ευθύμιος Αλέπης , Λέκτορας

Ημερομηνία Παράδοσης

Ιούνιος 2014

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Αλέπης Ευθύμιος

Τσιχριντζής Γεώργιος

Βίβρου Μαρία

Λέκτορας

Καθηγητής

Καθηγήτρια

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Πανεπιστήμιο Πειραιώς

**Αφιερωμένη σε όσους με
στήριξαν καθ' όλη τη διάρκεια
της προσπάθειας μου**



ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θεωρώ υποχρέωση μου να ευχαριστήσω όλους όσους συντέλεσαν με οποιοδήποτε τρόπο στην ολοκλήρωση αυτής της διπλωματικής εργασίας . Τα θερμά ευχαριστήριά μου θα ήθελα να απευθύνω στον επιβλέποντα καθηγητή μου Αλέπη Ευθύμιο για την πολύτιμη βοήθεια και καθοδήγηση του και την προτροπή να ασχοληθώ με αυτό το αντικείμενο. Επίσης , θερμά ευχαριστήρια απευθύνω και σε όλους τους καθηγητές μου στο Μεταπτυχιακό για την προσπάθεια που κατέβαλλαν να μας διδάξουν τα μυστικά της Πληροφορικής .

Τέλος , δεν θα μπορούσα να μην απευθύνω ένα μεγάλο ευχαριστώ στην οικογένεια , στην Κωνσταντίνα Παναγιωτίδου και τους στενούς μου φίλους και γνωστούς για την πολύτιμη υποστήριξη , ηθική και υλική , που απλόχερα μου προσέφεραν καθ' όλη την φοίτηση μου στο Μεταπτυχιακό Πρόγραμμα της Πληροφορικής.

Πανεπιστήμιο Πειραιώς

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έρχεται να ολοκληρώσει την φοίτηση μου στο Μεταπτυχιακό Πρόγραμμα της Πληροφορικής του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιά . Αντικείμενο μελέτης είναι η εκμάθηση και κατόπιν η υλοποίηση μιας εφαρμογής με τεχνολογία *Android*. Πιο συγκεκριμένα η εφαρμογή αυτή είχε ως θέμα τον προσδιορισμό ενός εβδομαδιαίου διαιτολογίου για τον χρήστη . Η τεχνολογία που χρησιμοποιήθηκε σε μεγαλύτερο βαθμό είναι *Java* , *XML* , *SQLite* και σε μικρότερο η *HTML* και η *JavaScript* . Χρησιμοποιήθηκαν εργαλεία όπως το *Eclipse* με το *Android sdk* , διάφοροι *text editors* και ένα κινητό τηλέφωνο *Samsung S3* για δοκιμές .

Η δομή της εφαρμογής είναι απλή . Σκοπός είναι ο χρήστης να δώσει πληροφορίες για τον εαυτό του στην εφαρμογή , όπως βάρος , ύψος , ηλικία , δραστηριότητα εντός και εκτός δουλειάς και η εφαρμογή να τις επεξεργαστεί και να του προτείνει ως απάντηση το εβδομαδιαίο διαιτολόγιο . Γίνεται χρήση διαιτολογικών τύπων για να βρεθούν παράμετροι σημαντικοί για τον προσδιορισμό της διαίτας (*BMI* , *BMR* , *activity factor* , *Daily Caloric Needs*) .

Τέλος ίσως ο βασικότερος σκοπός της εφαρμογής ήταν η εκμάθηση προγραμματισμού σε περιβάλλον *Android* . Αυτό επετεύχθη σε μεγάλο βαθμό καθώς χρησιμοποιήθηκαν πολλά διαφορετικά στοιχεία του *Android* , όπως *TextViews* , *ScrollViews* , *WebViews* , *EditTexts* , *AlertDialogs* , *ImageSlider* , *Spinners* , *Tabs* , *Calendar Api* , *Toast Messages* κ.α. Έχει δοθεί μεγάλη προσοχή και στο *user interface* της εφαρμογής ώστε να είναι ελκυστικό για τον χρήστη , σαν εφαρμογή στο *Android Market*.

The present thesis concludes my studies in the Master's Program on Informatics by the Department of Informatics of the University of Piraeus. The object of the study is the familiarization with and the implementation of an *Android* technology application. More precisely, the object of this application was to determine a weekly diet program for the user. The technologies which were mostly used include *Java*, *XML*, *SQLite* and, at a lesser degree, *HTML* and *JavaScript*. The project included the use of tools such as *Eclipse* and *Android sdk*, various text editors and a *Samsung S3* mobile phone for testing.

The structure of the application is simple. Its purpose is for the user to input data such as weight, height, age and rate of physical activity during work and leisure time, which the application can then process and, finally, suggest a weekly diet schedule. Moreover, nutritional types are used to provide parameters which are important for the determination of the diet (*BMI*, *BMR*, activity factor, *Daily Caloric Needs*).

Finally, possibly the most important goal of the application has been the familiarization with programming in an *Android* environment. This was largely accomplished as plenty of different *Android* elements like *TextViews*, *ScrollViews*, *WebViews*, *EditTexts*, *AlertDialogs*, *ImageSlider*, *Spinners*, *Tabs*, *Calendar Api*, *Toast Messages* e.t.c. were employed. Special attention has also been given to the application's user interface so that it will appeal to the user in a similar way as an application uploaded on *Android market* would.

Πίνακας περιεχομένων

ΚΕΦΑΛΑΙΟ 1^ο	8
1.1 ΕΙΣΑΓΩΓΗ	8
ΚΕΦΑΛΑΙΟ 2^ο	11
2.1 ANDROID	11
2.1.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ ANDROID	11
2.1.2 Η ΕΙΣΟΔΟΣ ΤΟΥ «ANDROID»	11
2.1.3 ΦΘΗΝΗ & ΕΥΚΟΛΗ ΑΝΑΠΤΥΞΗ	13
2.1.4 BETA VERSION	14
2.1.5 ANDROID 1.0	14
2.1.6 ANDROID 1.1	15
2.1.7 ANDROID 1.5 CUPCAKE.....	15
2.1.8 ANDROID 1.6 DONUT.....	16
2.1.9 ANDROID 2.0 ÉCLAIR.....	17
2.1.10 ANDROID 2.2 FROYO.....	18
2.1.11 ANDROID 2.3 GINGERBREAD.....	19
2.1.12 ANDROID 3.0 HONEYCOMB	21
2.1.13 ANDROID 4.0 ICE CREAM SANDWICH.....	23
2.1.14 ANDROID 4.1 - 4.2 - 4.3 JELLY BEAN	25
2.1.15 ANDROID 4.4 KITKAT	28
2.1.16 ΣΥΣΧΕΤΙΣΜΟΣ ANDROID ΕΚΔΟΣΕΩΝ - ΣΥΣΚΕΥΩΝ.....	29
3.1 ΔΙΑΙΤΑ	30
ΚΕΦΑΛΑΙΟ 4^ο	37
4.1 ΔΟΜΗ DIETDROID	37

4.2	ΑΡΧΕΙΑ ΓΡΑΦΙΚΩΝ & ΑΡΧΕΙΑ ΠΟΡΩΝ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ	38
4.3	ΑΜΕΤΑΓΛΩΤΤΙΣΤΑ ΑΡΧΕΙΑ ΠΟΡΩΝ	39
4.4	ΔΙΑΦΟΡΟΙ ΤΥΠΟΙ ΠΟΡΩΝ (ΧΡΩΜΑΤΑ , ΣΥΜΒΟΛΟΣΕΙΡΕΣ ΚΕΙΜΕΝΟΥ , ΣΤΥΛ , ΔΙΑΣΤΑΣΕΙΣ)	39
4.4.1	Colors.xml	39
4.4.2	Dimens.xml	39
4.4.3	Strings.xml	39
4.4.4	Styles.xml	39
4.5	ΑΚΑΤΕΡΓΑΣΤΑ ΑΡΧΕΙΑ ΠΟΡΩΝ	40
4.5.1	Format_loadinglayout.xml	40
4.5.2	Help_text.txt	40
ΚΕΦΑΛΑΙΟ 5^ο		41
5.1	ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΒΟΗΘΗΤΙΚΩΝ ΚΛΑΣΕΩΝ	41
5.1.1	Βοηθητικές Κλάσεις	41
5.1.2	ExitDroid.java	41
5.1.3	FullScreen.java	45
5.1.4	InputStreamToString.java	45
5.1.5	JumpToActivity.java	45
5.1.6	Prwina.java	45
5.1.7	Mesimeriana.java	46
5.1.8	Bradyna.java	46
5.1.9	DatabaseHandler.java	46
5.2	ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΚΥΡΙΩΝ ΚΛΑΣΕΩΝ	47
5.2.1	AndroidManifest.xml	49
5.2.2	loadinglayout.xml	51

5.2.3	LoadingActivity.java.....	52
5.2.4	menulayout.xml.....	60
5.2.5	menulayoutitem.xml	61
5.2.6	menu_mainmenu.xml.....	61
5.2.7	MenuActivity.java.....	62
5.2.8	helplayout.xml.....	72
5.2.9	HelpActivity.java.....	73
5.2.10	settinglayout.xml.....	79
5.2.11	SettingsActivity.java.....	82
5.2.12	finddietlayout.xml	90
5.2.13	input_data_layout.xml.....	92
5.2.14	calculator_data_layout.xml	98
5.2.15	spinner_item.xml.....	100
5.2.16	inputtab_spinner_borders.xml	101
5.2.17	inputtab_calculator_tab_edittext_borders.xml.....	101
5.2.18	tab_background_selectors.xml	102
5.2.19	tab_text_selectors.xml	102
5.2.20	FindDietActivity.java.....	102
5.2.21	resultslayout.xml.....	112
5.2.22	results_tab_layout.xml	114
5.2.23	ResultsActivity.java.....	116
ΚΕΦΑΛΑΙΟ 6^ο	121
6.1	ΣΥΜΠΕΡΑΣΜΑΤΑ & ΠΕΡΙΛΗΨΗ	121
6.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	122

ΚΕΦΑΛΑΙΟ 7^ο	122
7.1 ΒΙΒΛΙΟΓΡΑΦΙΑ	122
7.1.1 ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	122
7.1.2 ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	122
7.1.3 ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	123
ΚΕΦΑΛΑΙΟ 8^ο	123
8.1 ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΕΣ	123
8.1.1 Colors.xml	123
8.1.2 Dimens.xml	125
8.1.3 Strings.xml	126
8.1.4 Styles.xml	127
8.1.4 Format_loadinglayout.xml	130
8.1.5 ExitDroid.java	131
8.1.6 FullScreen.java	133
8.1.7 InputStreamToString.java	133
8.1.8 JumpToActivity.java	133
8.1.9 Prwina.java	134
8.1.10 Mesimeriana.java	135
8.1.11 Bradyna.java	137
8.1.12 DatabaseHandler.java	138
8.1.13 MenuActivity.java (τμήμα του)	142
8.1.14 FindDietActivity.java	148
8.1.15 ResultsActivity.java	169

Πίνακας περιεχομένων εικόνων

Εικόνα 1: Μερίδια Αγοράς & Εκδόσεις Λειτουργικών Συστημάτων (http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems).....	8
Εικόνα 2: Αριθμός Android Applications στο Google Play (http://www.appbrain.com/stats/number-of-android-apps).....	9
Εικόνα 3: Πωλήσεις Smartphones (http://www.kumulos.com/2013/05/08/kantar-backend-as-a-service)	10
Εικόνα 4: T-Mobile G1.....	12
Εικόνα 5: Android Figure.....	13
Εικόνα 6: Συσχετισμός Android Συσκευών - Εκδόσεων	30
Εικόνα 7: Exit Dialog.....	42
Εικόνα 8: Exit Dialog (Answer No)	43
Εικόνα 9: Exit Dialog (Answer Yes).....	44
Εικόνα 10: Διάγραμμα οθονών – δραστηριοτήτων.....	48
Εικόνα 11: loadinglayout.xml	52
Εικόνα 12: Loading Screen (1)	56
Εικόνα 13: Loading Screen (2)	57
Εικόνα 14: Loading Screen (3).....	58
Εικόνα 15: Loading Screen (4)	59
Εικόνα 16: menulayout.xml.....	62
Εικόνα 17: Menu Screen (1)	69
Εικόνα 18: Mobile Menu (1).....	70
Εικόνα 19: Mobile Menu(2).....	71
Εικόνα 20: helplayout.xml.....	73
Εικόνα 21: MenuScreen (2)	76
Εικόνα 22: Help Screen (1)	77

Εικόνα 23: Help Screen (2)	78
Εικόνα 24: settinglayout.xml	81
Εικόνα 25: Menu Screen (3)	88
Εικόνα 26: Settings Screen	89
Εικόνα 27: finddietlayout.xml	92
Εικόνα 28: input_data_layout.xml.....	97
Εικόνα 29: calculator_data_layout.xml.....	100
Εικόνα 30: Menu Screen (4)	105
Εικόνα 31: FindDiet Screen (1)	106
Εικόνα 32: FindDiet Screen (2)	107
Εικόνα 33: FindDiet Screen (3)	108
Εικόνα 34: FindDiet Screen (4)	109
Εικόνα 35: FindDiet Screen (5)	110
Εικόνα 36: FindDiet Screen (6)	111
Εικόνα 37: resultslayout.xml	113
Εικόνα 38: results_tab_layout.xml	115
Εικόνα 39: Results Screen (1)	117
Εικόνα 40: Results Screen (2)	118
Εικόνα 41: Calendar Screen (1).....	119
Εικόνα 42: Calendar Screen (2).....	120
Εικόνα 43: Top 10 Google Play categories	121

ΚΕΦΑΛΑΙΟ 1°

1.1 ΕΙΣΑΓΩΓΗ

Κάθε κινητό τηλέφωνο χρειάζεται ενός είδους λειτουργικό σύστημα για να λειτουργήσει και να αξιοποιήσει τις δυνατότητες που έχει να κάνει κλήσεις , να στέλνει μηνύματα , να τραβήξει φωτογραφίες κ.α. Τα παλαιότερα λειτουργικά συστήματα των κινητών ήταν πολύ απλά , μιας και οι δυνατότητες των τότε κινητών τηλεφώνων ήταν περιορισμένες. Όμως , στα μοντέρνα smartphones έχουν προστεθεί πολλά χαρακτηριστικά που προσεγγίζουν τα χαρακτηριστικά των σταθερών υπολογιστών , π.χ. *high-speed CPUs* , μεγάλοι σε χωρητικότητα δίσκοι , οθόνες *HD* , κάμερες , *wifi* , *Bluetooth* , κ.α. Έτσι έπρεπε να αναπτυχθούν λειτουργικά συστήματα που να μπορούν να υποστηρίξουν όλες αυτές τις δυνατότητες των νέων *smartphones*.

Πλέον υπάρχουν αρκετά λειτουργικά συστήματα για *mobile* συσκευές που προσπαθούν να κερδίσουν το δικό τους μερίδιο ο καθένας στην αγορά. Λειτουργικά όπως το iOS της *Apple* (≈13,4%), το *Firefox OS* της *Mozilla* (≈1,2%) , το *Windows Phone* της *Microsoft* (≈4,1%) και το *BlackBerry OS* της *BlackBerry* (≈1%) ανταγωνίζονται να αυξήσουν το μερίδιο τους και να κλέψουν κάτι από το μερίδιο του *Android OS* της *Google* (≈81,3%) . Αναλυτικότερα τα λειτουργικά συστήματα , τα μερίδια στη αγορά και οι εκδόσεις τους σήμερα:

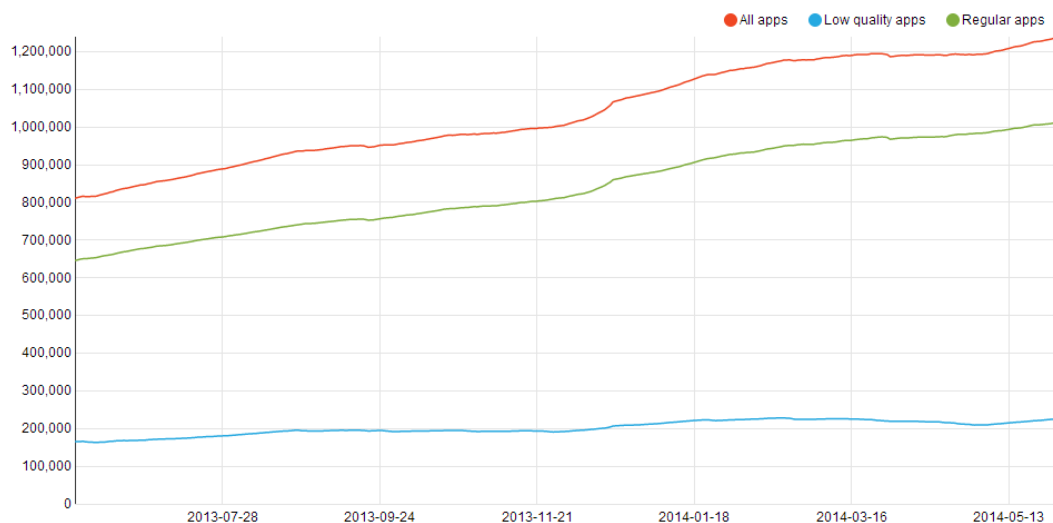
Feature	iOS	Android	Firefox OS	Windows Phone	BlackBerry OS	Series 40	Mer	Tizen	Sailfish OS	Ubuntu Touch
Company	Apple, Inc	Open Handset Alliance/Google	Open Web Device Compliance Review Board/Mozilla	Microsoft	BlackBerry Ltd.	Nokia	Linux Foundation	Linux Foundation, Tizen Association, Samsung, Intel	Sailfish Alliance, Jolla	Canonical Ltd. and Ubuntu community contributors
Market share ^[1]	13.4%	81.3%	N/A	4.1%	1.0%	N/A	<0.1%	N/A	N/A	
Current version	7.1.1	4.4.3	1.2 ^[2]	8.1	10.2.1	Developer Platform 2.0	1.2.0.10	2.2	1.0.5.19 Paarlampi	1.0

Εικόνα 1: Μερίδια Αγοράς & Εκδόσεις Λειτουργικών Συστημάτων
(http://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems)

Βλέποντας τον παραπάνω πίνακα μπορούμε να πούμε ότι το πράσινο «ανθρωπάκι» του *Android* βρίσκεται σχεδόν παντού . Αν και η πλατφόρμα *Android* αριθμεί λίγα μόνο χρόνια ζωής, έχει γίνει γρήγορα από τις πιο αγαπημένες μεταξύ των χρηστών. Η ομορφιά του *Android* είναι η ανοικτή πλατφόρμα που επιτρέπει την εύκολη δημιουργία εφαρμογών, η εύκολη πρόσβαση σε αρχεία πολυμέσων εν κινήσει και η μεγάλη ποικιλία των εφαρμογών που είναι διαθέσιμες, που τοποθετούν το λειτουργικό σύστημα *Android* σε μια πολύ ισχυρή θέση για το μέλλον. Η πρόσβαση σε χιλιάδες *Android* εφαρμογές και το φιλικό προς τον χρήστη περιβάλλον (*GUI*) είναι αυτό που κάνει το *Android OS* τόσο ελκυστικό στους χρήστες.

Η ραγδαία ανάπτυξη του *Android* φαίνεται και από τρομερή αύξηση του αριθμού των εφαρμογών που γίνονται διαθέσιμες για τους χρήστες καθημερινά στο *Google Play* . Πιο συγκεκριμένα τον Ιούλιο του 2013 υπήρχαν 888.563 εφαρμογές διαθέσιμες και το Μάιο του 2014 περίπου 1.207.586 . Άρα το *Android* σαν εργαλείο ανάπτυξης εφαρμογών είναι αποδεκτό και διαδεδομένο σε πολλούς developer ανά τον κόσμο .











Android apps on Google Play



Εικόνα 2: Αριθμός Android Applications στο Google Play
(<http://www.appbrain.com/stats/number-of-android-apps>)

Κάτι ανάλογο συμβαίνει και από την μεριά των χρηστών που προτιμούν να χρησιμοποιούν *Android smartphones*. Αυτό φαίνεται από τις πωλήσεις των κινητών αυτών σε διάφορες χώρες του κόσμου με στατιστικά στοιχεία του α' τριμήνου του 2013 π.χ. στη Γερμανία με 73,6%, στη Κίνα με 69,4% ακόμα και στην Αμερική την πατρίδα της *Apple* τα *Android smartphones* ξεπερνούν σε πωλήσεις εκείνα με *iOS*, 49,3% έναντι 43,7%.

Smartphone OS Sales Share (%)

Germany 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	USA 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	18.7	16.9	- 1.8	iOS	44.6	43.7	- 0.9
Android	63.4	73.6	10.2	Android	47.9	49.3	1.4
RIM	1.8	0.5	- 1.3	RIM	2.6	0.9	- 1.7
Symbian	8.2	2.5	- 5.7	Symbian	0.5	0.2	- 0.3
Windows	6.6	6.1	- 0.5	Windows	3.7	5.6	1.9
Other	1.2	0.4	- 0.8	Other	0.6	0.3	- 0.3
GB 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	China 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	30.1	28.7	- 1.4	iOS		24.6	
Android	49.3	58.4	9.1	Android		69.4	
RIM	15.3	5.1	- 10.2	RIM		0.3	
Symbian	2.1	0.6	- 1.5	Symbian		2.9	
Windows	2.9	7.0	4.1	Windows		2.0	
Other	0.2	0.3	0.1	Other		0.7	
France 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	Australia 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	20.5	21.2	0.7	iOS	33.8	31.0	- 2.8
Android	57.8	63.3	5.5	Android	52.9	61.7	8.8
RIM	7.4	4.0	- 3.4	RIM	0.8	0.5	- 0.3
Symbian	4.3	0.6	- 3.7	Symbian	6.7	1.1	- 5.6
Windows	2.8	7.2	4.4	Windows	3.3	4.1	0.8
Other	7.2	3.6	- 3.6	Other	2.6	1.5	- 1.1
Italy 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	Japan 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	22.6	19.9	- 2.7	iOS		49.2	
Android	48.4	62.5	14.1	Android		45.8	
RIM	3.7	2.5	- 1.2	RIM		0.7	
Symbian	17.7	3.5	- 14.2	Symbian		0.2	
Windows	5.9	10.9	5.0	Windows		0.3	
Other	1.6	0.7	- 0.9	Other		3.7	
Spain 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change	EUS 	3 m/e Mar 2012	3 m/e Mar 2013	% pt. Change
iOS	5.4	3.2	- 2.2	iOS	20.4	19.4	- 1.0
Android	76.8	93.5	16.7	Android	58.1	68.8	10.7
RIM	7.8	0.2	- 7.6	RIM	8.1	2.7	- 5.4
Symbian	8.1	1.5	- 6.6	Symbian	7.2	1.6	- 5.6
Windows	1.8	1.3	- 0.5	Windows	4.1	6.5	2.5
Other	0.0	0.3	0.3	Other	2.2	1.1	- 1.2

Εικόνα 3: Πωλήσεις Smartphones (<http://www.kumulos.com/2013/05/08/kantar-backend-as-a-service>)

Συμπέρασμα από όλα τα παραπάνω είναι ότι το *Android* είναι μέσα στη ζωή και την καθημερινότητα μας. Είναι το πλέον εύχρηστο λειτουργικό σύστημα, χρησιμοποιείται σε όλο και περισσότερα είδη συσκευών, π.χ. *Android TV*, *phablets*, *tablet*, *netbooks*, *smartbooks*, *smartglasses*, *smartwatches*, *headphones*, *car CD & DVD players* κ.α.. Όλα αυτά δείχνουν ότι παρότι το *Android* βρίσκεται κοντά μας από το 2007, αναπτύσσεται συνεχώς, είναι μια τεχνολογία αιχμής και θα μας απασχολεί και στο μέλλον. Έτσι αποτελεί μεγάλο προσόν για ένα *developer* να γνωρίζει την τεχνολογία του *Android* και πως μπορεί να δημιουργήσει εφαρμογές για την πλατφόρμα αυτή.

ΚΕΦΑΛΑΙΟ 2^ο

2.1 ANDROID

2.1.1 ΓΕΝΙΚΑ ΓΙΑ ΤΟ ANDROID

Το Android είναι η πρώτη *πλήρης, ανοικτή και δωρεάν διαθέσιμη* πλατφόρμα κινητών επικοινωνιών. Οι προγραμματιστές έχουν ένα πλήρες *kit* ανάπτυξης λογισμικού, με άφθονα εργαλεία για ανάπτυξη ισχυρών, πλούσιων σε χαρακτηριστικά εφαρμογών. Η πλατφόρμα είναι ανοικτού πηγαίου κώδικα, και βασίζεται σε δοκιμασμένα και σωστά πρότυπα ανοικτού πηγαίου κώδικα, με τα οποία οι προγραμματιστές είναι εξοικειωμένοι. Το καλύτερο απ' όλα είναι ότι δεν υπάρχουν ακριβοί φραγμοί εισόδου για τους προγραμματιστές και δεν απαιτούνται χρεώσεις για άδειες χρήσης. (Απαιτείται μία μικρή χρέωση για την δημοσίευση σε μηχανισμούς διανομής τρίτων, όπως είναι η αγορά του *Android*, «*Android Market*»). Οι προγραμματιστές του *Android* έχουν πολλές επιλογές για διανομή και εμπορική εκμετάλλευση των εφαρμογών τους.

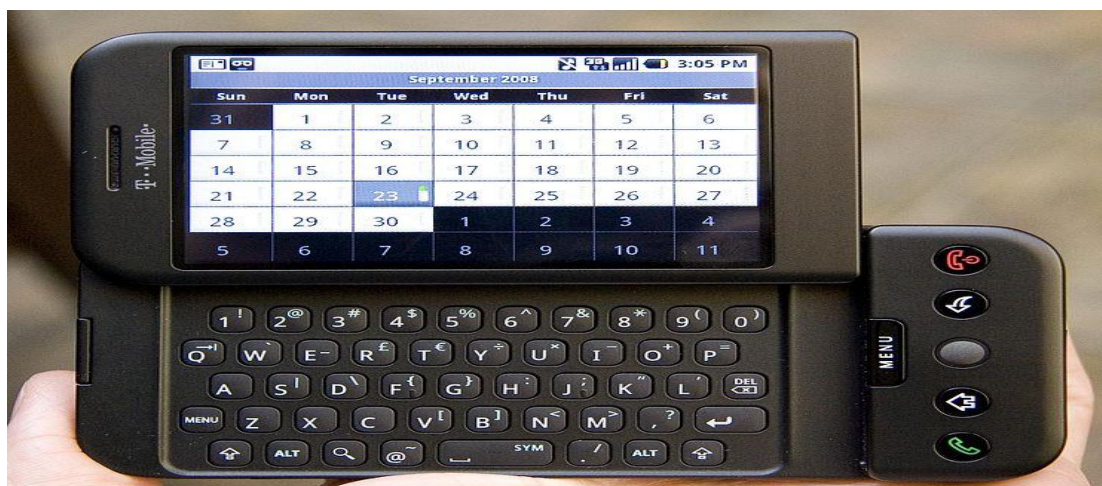
Το 2007, μία ομάδα κατασκευαστών τηλεφωνικών συσκευών, φορέων ασύρματων επικοινωνιών και προγραμματιστών λογισμικού (κυρίως η *Google*), σχημάτισαν την συμμαχία *Open Handset Alliance*, με στόχο την ανάπτυξη της επόμενης γενιάς πλατφόρμας ασύρματων επικοινωνιών. Σε αντίθεση με τις υπάρχουσες πλατφόρμες, αυτή η νέα πλατφόρμα δεν θα ήταν αποκλειστικής εκμετάλλευσης και θα βασιζόταν σε νέα ανοικτά πρότυπα, τα οποία θα οδηγούσαν σε χαμηλότερο κόστος ανάπτυξης και σε αυξημένα οφέλη. Οι προγραμματιστές λογισμικού για κινητές επικοινωνίες θα είχαν επίσης πλήρη πρόσβαση στα χαρακτηριστικά των τηλεφωνικών συσκευών, επιτρέποντας να επιτυγχάνεται μεγαλύτερη καινοτομία.

Ενώ οι πλατφόρμες αποκλειστικής εκμετάλλευσης όπως οι *RIM BlackBerry* και *Apple iPhone* κέρδιζαν έδαφος, η κοινότητα των προγραμματιστών ανάπτυξης για κινητές επικοινωνίες άκουσαν με χαρά τα νέα γι' αυτήν την πλατφόρμα, που θα άλλαζε τους όρους του παιχνιδιού.

2.1.2 Η ΕΙΣΟΔΟΣ ΤΟΥ «ANDROID»

Το 2007, η *Open Handset Alliance* ανακοίνωσε την πλατφόρμα *Android* και εκκίνησε ένα πρόγραμμα «*beta*» για προγραμματιστές. Το *Android* πέρασε απ' την τυπική διαδικασία αναθεωρήσεων μιας νέας πλατφόρμας. Εκδόθηκαν αρκετές αναθεωρήσεις προ της τελικής έκδοσης του *Android Software Development Kit 1.1 (SDK)*. Η πρώτη τηλεφωνική συσκευή *Android (T-Mobile G1)* άρχισε να διανέμεται στα τέλη του 2008.

The **HTC Dream** (also known as the **T-Mobile G1** in the United States and parts of Europe, and as the **Era G1** in Poland) is a smartphone developed by HTC. First released in October 2008, the Dream was the first commercially released device to use the Linux-based Android operating system (βλ. wikipedia/HTC Dream)



Εικόνα 4: T-Mobile G1

Το 2009 και το 2010, νέα και εντυπωσιακά *smartphones* *Android* έφτασαν στις αγορές σ' όλο τον κόσμο και η πλατφόρμα απέδειξε τις δυνατότητές της στην βιομηχανία και στους καταναλωτές. Τα τρία τελευταία χρόνια έχουν γίνει αρκετές αναθεωρήσεις της πλατφόρμας *Android*, όπου η καθεμία παρέχει χρήσιμα χαρακτηριστικά, τα οποία χρησιμοποιούν οι προγραμματιστές και απολαμβάνουν οι χρήστες. Πρόσφατα, οι πλατφόρμες κινητών επικοινωνιών έχουν αρχίσει να χρησιμοποιούνται σε άλλες συσκευές πέρα απ' τα παραδοσιακά *smartphones*, όπως είναι οι ταμπλέτες, τα συστήματα ανάγνωσης ηλεκτρονικών βιβλίων και συσκευές όπως η *Google TV*.

Μέχρι τώρα υπάρχουν εκατοντάδες ποικιλίες συσκευών *Android* στις αγορές σε όλο τον κόσμο -- από τα υψηλής τεχνολογίας *smartphones* μέχρι τις φθηνές τηλεφωνικές συσκευές και οτιδήποτε ανάμεσα στα δύο αυτά άκρα. Αυτός ο αριθμός δεν περιλαμβάνει τις αναρίθμητες ταμπλέτες και συστήματα ανάγνωσης ηλεκτρονικών βιβλίων που διατίθενται επίσης, τις δεκάδες νέες συσκευές που έχουν ανακοινωθεί επίσης, ή τα ηλεκτρονικά καταναλωτικά προϊόντα που εκτελούν το *Android*. Υπάρχουν περισσότερες από 1.000.000 εφαρμογές που έχουν εκδοθεί στην *Android Market* (καταγεγραμμένες μέχρι Ιούλιος 2013). Στις Η.Π.Α. όλοι οι κύριοι φορείς έχουν σήμερα στις γραμμές παραγωγής τους τηλεφωνικές συσκευές *Android*, και το ίδιο συμβαίνει στην Ασία, στην Ευρώπη, στην Κεντρική και στην Νότια Αμερική και αλλού. Ο ρυθμός των νέων συσκευών *Android* που φτάνουν στις αγορές όλου του κόσμου συνεχίζει να αυξάνεται.

Google Play, formerly the **Android Market**, is a digital distribution platform for applications for the Android operating system and an online electronics and digital media store, operated by Google. The service allows users to browse and download applications developed with the Android SDK and published through Google, as well as music, magazines, books, movies, and television programs. Users can also purchase hardware, such as Chromebooks Google Nexus-branded mobile devices, Chromecasts, and accessories, through Google Play. (βλ. wikipedia/GooglePlay)

Η *Google* ήταν ένα μέλος με μεγάλη συνεισφορά της *Open Handset Alliance* από την αρχή της. Η εταιρεία φιλοξενεί το έργο ανοικτού πηγαίου κώδικα τον *Android*, καθώς και το *site* του προγραμματιστή, στην <http://developer.android.com>. Μπορείτε να επισκεφθείτε αυτό το *site* για να φορτώσετε το *Android SDK*, να πάρετε την πιο πρόσφατη τεκμηρίωση πλατφόρμας και να ψάξετε μέσα στα *forum* των προγραμματιστών του *Android*. Η *Google* λειτουργεί επίσης την δημοφιλέστερη υπηρεσία για πώληση εφαρμογών *Android* σε τελικούς χρήστες, την *Android Market*. Η μασκότ του *Android* είναι το μικρό πράσινο ρομπότ που φαίνεται στην παρακάτω εικόνα.

Ανάπτυξη *Android* Εφαρμογής «Δίατα (DietDroid)»



Εικόνα 5: Android Figure

2.1.3 ΦΘΗΝΗ & ΕΥΚΟΛΗ ΑΝΑΠΤΥΞΗ

Πλέον η ανάπτυξη εφαρμογών για κινητές επικοινωνίες γίνεται με εύκολο και απλό τρόπο. Αντίθετα η ανάπτυξη εφαρμογών για ασύρματες επικοινωνίες, με τους πάρα πολύ ακριβούς μεταγλωττιστές και τα προνομιακά προγράμματα προγραμματιστών, είναι πολύ ακριβή για να μπορεί να χρησιμοποιηθεί σε επιτραπέζιους υπολογιστές. Εδώ, το *Android* σπάει το μοντέλο αποκλειστικής εκμετάλλευσης. Σε με τις άλλες πλατφόρμες κινητών επικοινωνιών, στην πραγματικότητα δεν υπάρχει κόστος για ανάπτυξη εφαρμογών για το *Android*.

Το *Android SDK* και τα εργαλεία του *Android* διατίθενται δωρεάν στο site του προγραμματιστή του *Android*, <http://developer.android.com>. Το δωρεάν διατιθέμενο πρόγραμμα *Eclipse* έγινε το δημοφιλέστερο ολοκληρωμένο περιβάλλον ανάπτυξης (*Integrated Development Environment, IDE*) για ανάπτυξη εφαρμογών *Android*.

Οι εφαρμογές *Android* γράφονται σε *Java*, μία απ' τις δημοφιλέστερες γλώσσες προγραμματισμού στον κόσμο. Οι προγραμματιστές της *Java* θα είναι εξοικειωμένοι με πολλά από τα πακέτα, που παρέχονται ως τμήμα του *Android SDK*. Έτσι κάποιος προγραμματιστής μπορεί να χρησιμοποιήσει το δημοφιλέστερο περιβάλλον ανάπτυξης, το *Eclipse*, που είναι δωρεάν και υποστηρίζεται επαρκώς από την ομάδα του *Android*, μπορεί να γράφει τις εντολές στο συνηθέστερο λειτουργικό σύστημα *Windows* και μπορεί να επιλέξει για ποιες συσκευές θα αναπτύξει την εφαρμογή του (με βάση την έκδοση του *Android*, στην οποία τρέχουν οι συσκευές).

2.1.4 BETA VERSION

Η επίσημη πορεία της πλατφόρμας του *Android* αρχίζει με την έκδοση *beta*. Πριν από αυτό είχαν δημιουργηθεί τουλάχιστον 2 μη εμπορικές προεκτάσεις, που τις λειτουργούσαν εσωτερικά της *Google* και της *Open Handset Alliance*. Για τις εκδόσεις αυτές είχαν χρησιμοποιηθεί και πρότυπα ονόματα από ήρωες κινουμένων σχεδίων, «*Astro Boy*», «*Bender*», «*R2-D2*». Οι επόμενες εκδόσεις της πλατφόρμας χωρίζονται χρονολογικά ανάλογα με το «*application programming interface (API)*» τους. Ο *DAN Morril* δημιούργησε κάποια από τα πρώτα *logo* της πλατφόρμας, όμως αυτό που επικράτησε ήταν της *Irina Block*. Ο *project manager Ryan Gibson* συνέλαβε την ιδέα του συνδυασμού της μασκώτ με γλυκίσματα για την ονομασία των επόμενων εκδόσεων του *Android*, πράγμα που εφαρμόστηκε από την έκδοση 1.5 και έπειτα. Η *beta version* κυκλοφόρησε στις 5 Νοεμβρίου 2007 και το αντίστοιχο *SDK* της στις 12 Νοεμβρίου 2007. Είχαν κυκλοφορήσει οι παρακάτω εκδόσεις του *SDK*:

- 16 November 2007: m3-rc22a
- 14 December 2007: m3-rc37a
- 13 February 2008: m5-rc14
- 3 March 2008: m5-rc15
- 18 August 2008: 0.9
- 23 September 2008: 1.0-r1

2.1.5 ANDROID 1.0

Η πρώτη εμπορική επίσημη απόπειρα της πλατφόρμας κυκλοφόρησε στις 23 Σεπτεμβρίου 2008. Συνδυάστηκε με την πρώτη επίσημη εμπορική συσκευή για την πλατφόρμα του *Android*, την *HTC Dream*. Υποστήριζε τα ακόλουθα:

Android 1.0 (API level 1) (βλ. wikipedia/AndroidVersions)
Android Market application download and updates through the Market application
Web browser to show, zoom and pan full HTML and XHTML web pages – multiple pages show as windows ("cards")
Camera support – however, this version lacked the option to change the camera's resolution, white balance, quality, etc.
Folders allowing the grouping of a number of application icons into a single folder icon on the Home screen
Access to web email servers, supporting POP3, IMAP4, and SMTP
Gmail synchronization with the Gmail application
Google Contacts synchronization with the People application
Google Calendar synchronization with the Calendar application
Google Maps with Street View to view maps and satellite imagery, as well as find local business and obtain driving directions using GPS
Google Sync, allowing management of over-the-air synchronization of Gmail, People, and Calendar
Google Search, allowing users to search the Internet and phone applications, contacts, calendar, etc.

Google Talk instant messaging
Instant messaging, text messaging, and MMS
Media Player, enabling management, importing, and playback of media files – however, this version lacked video and stereo Bluetooth support
Notifications appear in the Status bar, with options to set ringtone, LED or vibration alerts
Voice Dialer allows dialing and placing of phone calls without typing a name or number
Wallpaper allows the user to set the background image or photo behind the Home screen icons and widgets
YouTube video player
Other applications include: Alarm Clock, Calculator, Dialer (Phone), Home screen (Launcher), Pictures (Gallery), and Settings
Wi-Fi and Bluetooth support

2.1.6 ANDROID 1.1


Η έκδοση αυτή κυκλοφόρησε στις 9 Φεβρουαρίου του 2009 , αρχικά για το *HTC Dream* . Ήταν γνωστή με το όνομα « *Petit Four* » , ένα όνομα που δεν χρησιμοποιήθηκε ποτέ επίσημα. Σε σχέση με την προηγούμενη έκδοση προσέθετε χαρακτηριστικά, έλυνε προβλήματα και άλλαξε το *API* . Είχε τα παρακάτω επιπλέον χαρακτηριστικά:

Android 1.1 (API level 2) (βλ. wikipedia/AndroidVersions)
Details and reviews available when a user searches for businesses on Maps
Longer in-call screen timeout default when using the speakerphone, plus ability to show/hide dialpad
Ability to save attachments in messages
Support added for marquee in system layouts

2.1.7 ANDROID 1.5 CUPCAKE


Στις 27 Απριλίου 2009 κυκλοφόρησε η έκδοση 1.5 της πλατφόρμας. Ήταν βασισμένη στο πυρήνα 2.6.27 του *Linux* . Ήταν η πρώτη έκδοση που χρησιμοποίησε ένα κωδικό όνομα (*Cupcake*) και προσέφερε αρκετά νέα χαρακτηριστικά , που είναι :

Android 1.5 (API level 3) (βλ. wikipedia/AndroidVersions)
Support for third-party virtual keyboards with text prediction and user dictionary for custom words
Support for Widgets – miniature application views that can be embedded in other applications

(such as the Home screen) and receive periodic updates	
Video recording and playback in MPEG-4 and 3GP formats	
Auto-pairing and stereo support for Bluetooth (A2DP and AVRCP profiles)	
Copy and paste features in web browser	
User pictures shown for Favorites in Contacts	
Specific date/time stamp shown for events in call log, and one-touch access to a contact card from call log event	
Animated screen transitions	
Auto-rotation option	
New stock boot animation	
Ability to upload videos to YouTube	
Ability to upload photos to Picasa	


2.1.8 ANDROID 1.6 DONUT

Στις 15 Σεπτεμβρίου 2009 κυκλοφόρησε η έκδοση 1.6 της πλατφόρμας με το κωδικό όνομα *Donuts* , βασισμένο στον 2.6.29 πυρήνα του *Linux* . Τα επιπλέον χαρακτηριστικά του είναι:

Android 1.6 (API level 4) (βλ. wikipedia/AndroidVersions)	
Voice and text entry search enhanced to include bookmark history, contacts, and the web	
Ability for developers to include their content in search results	
Multi-lingual speech synthesis engine to allow any Android application to "speak" a string of text	
Easier searching and ability to view app screenshots in Android Market	
Gallery, camera and camcorder more fully integrated, with faster camera access	
Ability for users to select multiple photos for deletion	
Updated technology support for CDMA/EVDO, 802.1x, VPNs, and a text-to-speech engine	
Support for WVGA screen resolutions	
Speed improvements in searching and camera applications	
Expanded Gesture framework and new GestureBuilder development tool	

2.1.9 ANDROID 2.0 ÉCLAIR

Στις 26 Οκτωβρίου 2009 κυκλοφόρησε η επομένη έκδοση της πλατφόρμας με κωδικό όνομα « *Eclair* ». Επίσης κυκλοφόρησαν και συμπληρωματικές εκδόσεις του « *Eclair* », η 2.0.1 (API Level 6) στις 3 Δεκεμβρίου 2009 και η 2.1 (API Level 7) στις 12 Ιανουαρίου 2010. Όλες βασίστηκαν στο πυρήνα 2.6.29 του *Linux* και περιλάμβαναν τα ακόλουθα:

Android 2.0 (API level 5) (βλ. wikipedia/AndroidVersions)	
Expanded Account sync, allowing users to add multiple accounts to a device for synchronization of email and contacts	
Microsoft Exchange email support, with combined inbox to browse email from multiple accounts in one page	
Bluetooth 2.1 support	
Ability to tap a Contacts photo and select to call, SMS, or email the person	
Ability to search all saved SMS and MMS messages, with delete oldest messages in a conversation automatically deleted when a defined limit is reached	
Numerous new camera features, including flash support, digital zoom, scene mode, white balance, color effect and macro focus	
Improved typing speed on virtual keyboard, with smarter dictionary that learns from word usage and includes contact names as suggestions	
Refreshed browser UI with bookmark thumbnails, double-tap zoom and support for HTML5	
Calendar agenda view enhanced, showing attending status for each invitee, and ability to invite new guests to events	
Addition of live wallpapers, allowing the animation of home-screen background images to show movement	
Support for more screen sizes and resolutions, with better contrast ratio	
Improved Google Maps 3.1.2	
MotionEvent class enhanced to track multi-touch events	
Optimized hardware speed and revamped UI	
Minor API changes, bug fixes and framework behavioral changes	
Android 2.0.1 (API level 6) (βλ. wikipedia/AndroidVersions)	

Android 2.1 (API level 7) (βλ. wikipedia/AndroidVersions)

Minor amendments to the API and bug fixes

2.1.10 ANDROID 2.2 FROYO

Η επόμενη βασική έκδοση του *Android* ήταν η 2.2 με κωδικό όνομα «*Froyo*». Εκδόθηκε στις 20 Μαΐου 2010 και την ακολούθησαν οι συμπληρωματικές εκδόσεις 2.2.1 στις 18 Ιανουαρίου 2011 , 2.2.2 στις 22 Ιανουαρίου 2011 και 2.2.3 στις 21 Νοεμβρίου 2011. Τα χαρακτηριστικά τους ήταν τα ακόλουθα:

Android 2.2 (API level 8) (βλ. wikipedia/AndroidVersions)

Speed, memory, and performance optimizations

Additional application speed improvements, implemented through JIT compilation

Integration of Chrome's V8 JavaScript engine into the Browser application

Support for the Android Cloud to Device Messaging (C2DM) service, enabling push notifications

Improved Microsoft Exchange support, including security policies, auto-discovery, GAL look-up, calendar synchronization and remote wipe

Improved application launcher with shortcuts to Phone and Browser applications

USB tethering and Wi-Fi hotspot functionality

Option to disable data access over mobile network

Updated Market application with batch and automatic update features

Quick switching between multiple keyboard languages and their dictionaries

Support for Bluetooth-enabled car and desk docks

Support for numeric and alphanumeric passwords

Support for file upload fields in the Browser application

Support for installing applications to the expandable memory

Adobe Flash support



The browser now shows all frames of animated GIFs instead of just the first frame only

Support for high-PPI displays (up to 320 ppi), such as 4" 720p screens

Gallery allows users to view picture stacks using a zoom gesture


Android 2.2.1 (API level 8) (βλ. wikipedia/AndroidVersions)

Bug fixes, security updates and performance improvements
Android 2.2.2 (API level 8) (βλ. wikipedia/AndroidVersions)
Minor bug fixes, including SMS routing issues that affected the Nexus One
Android 2.2.3 (API level 8) (βλ. wikipedia/AndroidVersions)
Two security patches

2.1.11 ANDROID 2.3 GINGERBREAD


Η επόμενη βασική έκδοση του *Android* ήταν η 2.3 με κωδικό όνομα «*Gingerbread*». Εκδόθηκε στις 6 Δεκεμβρίου 2010 και την ακολούθησαν οι συμπληρωματικές εκδόσεις 2.3.1 το Δεκέμβρη του 2010, η 2.3.2 τον Ιανουάριο 2011, η 2.3.3 στις 9 Φεβρουαρίου του 2011, η 2.3.4 στις 28 Απριλίου του 2011, η 2.3.5 στις 25 Ιουλίου 2011, η 2.3.6 στις 2 Σεπτεμβρίου του 2011 και η 2.3.7 στις 21 Σεπτεμβρίου του 2011. Βασίστηκαν στον 2.6.35 πυρήνα του *Linux*. Τα χαρακτηριστικά τους ήταν τα ακόλουθα:

Android 2.3 (API level 9) (βλ. wikipedia/AndroidVersions)
Updated user interface design with increased simplicity and speed
Support for extra-large screen sizes and resolutions (WXGA and higher)
Native support for SIP VoIP internet telephony
Faster, more intuitive text input in virtual keyboard, with improved accuracy, better suggested text and voice input mode
Enhanced copy/paste functionality, allowing users to select a word by press-hold, copy, and paste
Support for Near Field Communication (NFC), allowing the user to read an NFC tag embedded in a poster, sticker, or advertisement
New audio effects such as reverb, equalization, headphone virtualization, and bass boost
New Download Manager, giving users easy access to any file downloaded from the browser, email, or another application
Support for multiple cameras on the device, including a front-facing camera, if available
Support for WebM/VP8 video playback, and AAC audio encoding
Improved power management with a more active role in managing applications that are keeping the device awake for too long
Enhanced support for native code development
Switched from YAFFS to ext4 on newer devices
Audio, graphical, and input enhancements for game developers

Concurrent garbage collection for increased performance	
Native support for more sensors (such as gyroscopes and barometers)	
Android 2.3.1 (API level 9) (βλ. wikipedia/AndroidVersions)	
Improvements and bug fixes for the Google Nexus S	
Android 2.3.2 (API level 9) (βλ. wikipedia/AndroidVersions)	
Improvements and bug fixes for the Google Nexus S	
Android 2.3.3 (API level 10) (βλ. wikipedia/AndroidVersions)	
Several improvements and API fixes	
Android 2.3.4 (API level 10) (βλ. wikipedia/AndroidVersions)	
Open Accessory Library support. Open Accessory was introduced in 3.1 (Honeycomb) but the Open Accessory Library grants 2.3.4 added support when connecting to a USB peripheral with compatible software and a compatible application on the device	
Switched the default encryption for SSL from AES256-SHA to RC4-MD5	
Support for voice or video chat using Google Talk	
Android 2.3.5 (API level 10) (βλ. wikipedia/AndroidVersions)	
Improved network performance for the Nexus S 4G, among other fixes and improvements	
Fixed Bluetooth bug on Samsung Galaxy S	
Improved Gmail application	
Shadow animations for list scrolling	
Camera software enhancements	
Improved battery efficiency	
Android 2.3.6 (API level 10) (βλ. wikipedia/AndroidVersions)	
Fixed a voice search bug	
The 2.3.6 update had the side-effect of impairing the Wi-Fi hotspot functionality of many Canadian Nexus S phones. Google acknowledged this problem and fixed it in late September	
Android 2.3.7 (API level 10) (βλ. wikipedia/AndroidVersions)	
Google Wallet support for the Nexus S 4G	

2.1.12 ANDROID 3.0 HONEYCOMB

Η επόμενη βασική έκδοση του *Android* ήταν η 3.0 με κωδικό όνομα «*Honeycomb*». Εκδόθηκε στις 22 Φεβρουαρίου 2011 και την ακολούθησαν οι συμπληρωματικές εκδόσεις 3.1 στις 10 Μαΐου του 2011, η 3.2 στις 15 Ιουλίου 2011, η 3.2.1 στις 20 Σεπτεμβρίου του 2011, η 3.2.2 στις 30 Αυγούστου του 2011, η 3.2.3, η 3.2.4 το Δεκέμβριο του 2011, η 3.2.5 τον Ιανουάριο του 2012 και η 3.2.6 το Φεβρουάριο του 2012. Βασίστηκαν στον 2.6.36 πυρήνα του *Linux*. Ήταν η πρώτη έκδοση του *Android* που κυκλοφόρησε για tablet. Τα χαρακτηριστικά τους ήταν τα ακόλουθα:

Android 3.0 (API level 11) (βλ. wikipedia/AndroidVersions)	
Optimized tablet support with a new “holographic” user interface	
Added System Bar, featuring quick access to notifications, status, and soft navigation buttons, available at the bottom of the screen	
Added Action Bar, giving access to contextual options, navigation, widgets, or other types of content at the top of the screen	
Simplified multitasking – tapping Recent Applications in the System Bar allows users to see snapshots of the tasks underway and quickly jump from one application to another	
Redesigned keyboard, making typing fast, efficient and accurate on larger screen sizes	
Simplified, more intuitive copy/paste interface	
Multiple browser tabs replacing browser windows, plus form auto-fill and a new “incognito” mode allowing anonymous browsing	
Quick access to camera exposure, focus, flash, zoom, front-facing camera, time-lapse, and other camera features	
Ability to view albums and other collections in full-screen mode in Gallery, with easy access to thumbnails for other photos	
New two-pane Contacts UI and Fast Scroll to let users easily organize and locate contacts	
New two-pane Email UI to make viewing and organizing messages more efficient, allowing users to select one or more messages	
Hardware acceleration	
Support for multi-core processors	
Ability to encrypt all user data	
HTTPS stack improved with Server Name Indication (SNI)	
Filesystem in Userspace (FUSE; kernel module)	
Applications' write access to secondary storage (memory cards on devices with internal primary storage) is disabled, while full access to internal primary storage is still allowed through a separate application-level permission	


Android 3.1 (API level 12) (βλ. wikipedia/AndroidVersions)
UI refinements
Connectivity for USB accessories
Expanded Recent Applications list
Resizable Home screen widgets
Support for external keyboards and pointing devices
Support for joysticks and gamepads
Support for FLAC audio playback
High-performance Wi-Fi lock, maintaining high-performance Wi-Fi connections when device screen is off
Support for HTTP proxy for each connected Wi-Fi access point
Android 3.2 (API level 13) (βλ. wikipedia/AndroidVersions)
Improved hardware support, including optimizations for a wider range of tablets
Increased ability of applications to access files on the SD card, e.g. for synchronization
Compatibility display mode for applications that have not been optimized for tablet screen resolutions
New display support functions, giving developers more control over display appearance on different Android devices
Android 3.2.1(API level 13) (βλ. wikipedia/AndroidVersions)
Bug fixes and minor security, stability and Wi-Fi improvements
Update to Android Market with automatic updates and easier-to-read Terms and Conditions text
Update to Google Books
Improved Adobe Flash support in browser
Improved Chinese handwriting prediction
Android 3.2.2 (API level 13) (βλ. wikipedia/AndroidVersions)
Bug fixes and other minor improvements for the Motorola Xoom 4G
Android 3.2.3 (API level 13) (βλ. wikipedia/AndroidVersions)
Bug fixes and other minor improvements for the Motorola Xoom and Motorola Xoom 4G
Android 3.2.4 (API level 13) (βλ. wikipedia/AndroidVersions)

"Pay as You Go" support for 3G and 4G tablets
Android 3.2.5(API level 13) (βλ. wikipedia/AndroidVersions)
Bug fixes and other minor improvements for the Motorola Xoom and Motorola Xoom 4G
Android 3.2.6(API level 13) (βλ. wikipedia/AndroidVersions)
Fixed data connectivity issues when coming out of airplane mode on the US 4G Motorola Xoom

2.1.13 ANDROID 4.0 ICE CREAM SANDWICH

Η επόμενη βασική έκδοση του *Android* ήταν η 4.0 με κωδικό όνομα «*Ice Cream Sandwich*». Εκδόθηκε στις 19 Οκτωβρίου 2011 και την ακολούθησαν οι συμπληρωματικές εκδόσεις 4.0.1 στις 21 Οκτωβρίου του 2011, η 4.0.2 στις 28 Νοεμβρίου 2011, η 4.0.3 στις 16 Δεκεμβρίου του 2011 και η 4.0.4 στις 29 Μαρτίου του 2012. Βασίστηκαν στον 3.0.1 πυρήνα του *Linux*. Τα χαρακτηριστικά τους ήταν τα ακόλουθα:

Android 4.0 (API level 14) (βλ. wikipedia/AndroidVersions)
Major refinements to the "Holo" interface with new Roboto font family
Soft buttons from Android 3.x are now available for use on phones
Separation of widgets in a new tab, listed in a similar manner to applications
Easier-to-create folders, with a drag-and-drop style
A customizable launcher
Improved visual voicemail with the ability to speed up or slow down voicemail messages
Pinch-to-zoom functionality in the calendar
Integrated screenshot capture (accomplished by holding down the Power and Volume-Down buttons)
Improved error correction on the keyboard
Ability to access applications directly from lock screen
Improved copy and paste functionality
Better voice integration and continuous, real-time speech to text dictation
Face Unlock, a feature that allows users to unlock handsets using facial recognition software
Automatic syncing of browser with users' Chrome bookmarks
Data Usage section in settings that lets users set warnings when they approach a certain usage limit, and disable data use when the limit is exceeded

Ability to shut down applications that are using data in the background	
Improved camera application with zero shutter lag, time lapse settings, panorama mode, and the ability to zoom while recording	
Built-in photo editor	
person Android Beam, a near-field communication feature allowing the rapid short-range exchange of web bookmarks, contact info, directions, YouTube videos and other data	
Refreshed "People" application with social network integration, status updates and hi-res images	
New gallery layout, organized by location and	
Support for the WebP image format	
Hardware acceleration of the UI	
Wi-Fi Direct	
1080p video recording for stock Android devices	
Android VPN Framework (AVF), and TUN (but not TAP) kernel module. Prior to 4.0, VPN software required rooted Android	
Android 4.0.1 (API level 14) (βλ. wikipedia/AndroidVersions)	
Fixed minor bugs for the Samsung Galaxy Nexus	
Android 4.0.2 (API level 14) (βλ. wikipedia/AndroidVersions)	
Fixed minor bugs on the Verizon Galaxy Nexus, the US launch of which was later delayed until December 2011	
Android 4.0.3(API level 14) (βλ. wikipedia/AndroidVersions)	
Numerous bug fixes and optimizations	
Improvements to graphics, databases, spell-checking and Bluetooth functionality	
New APIs for developers, including a social stream API in the Contacts provider	
Calendar provider enhancements	
New camera applications enhancing video stabilization and QVGA resolution	
Accessibility refinements such as improved content access for screen readers	
Android 4.0.4 (API level 14) (βλ. wikipedia/AndroidVersions)	
Stability improvements	


Better camera performance
Smoother screen rotation
Improved phone number recognition

2.1.14 ANDROID 4.1 - 4.2 - 4.3 JELLY BEAN

Η επόμενη βασική έκδοση του *Android* ήταν η 4.1 με κωδικό όνομα «*Jelly Bean*». Εκδόθηκε στις 9 Ιουλίου 2012 και την ακολούθησαν οι συμπληρωματικές εκδόσεις 4.1.1 στις 23 Ιουλίου 2012, η 4.1.2 στις 9 Οκτωβρίου 2012, η 4.2 στις 13 Νοεμβρίου 2012, η 4.2.1 στις 27 Νοεμβρίου του 2012, η 4.2.2 στις 11 Φεβρουαρίου του 2013, η 4.3 στις 24 Ιουλίου του 2013 και η 4.3.1 στις 3 Οκτωβρίου του 2013. Βασίστηκαν στους 3.0.31 και 3.4.0 πυρήνες του *Linux*. Τα χαρακτηριστικά τους ήταν τα ακόλουθα:

Android 4.1 (API level 16) (βλ. wikipedia/AndroidVersions)
Vsync timing across all drawing and animation done by the Android framework, including application rendering, touch events, screen composition and display refresh
Triple buffering in the graphics pipeline
Enhanced accessibility
Bi-directional text and other language support
User-installable keyboard maps
Expandable notifications
Ability to turn off notifications on an application-specific basis
Shortcuts and widgets can automatically be re-arranged or re-sized to allow new items to fit on home screens
Bluetooth data transfer for Android Beam
Offline voice dictation
Improved voice search
Tablets with smaller screens now use an expanded version of the interface layout and home screen used by phones
Improved camera application
Multichannel audio
USB audio (for external sound DACs)
Audio chaining (also known as gapless playback)
Ability for other launchers to add widgets from the application drawer without requiring root

access
Android 4.1.1 (API level 16) (βλ. wikipedia/AndroidVersions)
Fixed a bug on the Nexus 7 regarding the inability to change screen orientation in any application
Android 4.1.2 (API level 16) (βλ. wikipedia/AndroidVersions)
Lock/home screen rotation support for the Nexus 7
One-finger gestures to expand/collapse notifications
Bug fixes and performance enhancements
Android 4.2(API level 17) (βλ. wikipedia/AndroidVersions)
Lock screen improvements, including widget support and the ability to swipe directly to camera
Notification power controls ("Quick Settings")
"Daydream" screensavers, showing information when idle or docked
Multiple user accounts (tablets only)
Support for wireless display (Miracast)
Accessibility improvements: triple-tap to magnify the entire screen, pan and zoom with two fingers. Speech output and Gesture Mode navigation for blind users
New clock application with built-in world clock, stop watch and timer
All devices now use the same interface layout, previously adapted from phones on 4.1 for smaller tablets (with centered software buttons, the system bar at the top of the screen, and a home screen with a dock and centered application menu), regardless of screen size
Increased number of extended notifications and Actionable Notifications for more applications, allowing users to respond to certain notifications within the notification bar and without launching the application directly
SELinux
Always-on VPN
Premium SMS confirmation
Group Messaging
Android 4.2.1 (API level 17) (βλ. wikipedia/AndroidVersions)
Fixed a bug in the People application where December was not displayed on the date selector when adding an event to a contact
Added Bluetooth gamepads and joysticks as supported HID (Human interface device)

Android 4.2.2 (API level 17) (βλ. wikipedia/AndroidVersions)	
Fixed Bluetooth audio streaming bugs	
Long-pressing the Wi-Fi and Bluetooth icons in Quick Settings now toggles the on/off state	
New download notifications, which now shows the percentage and estimated time remaining for active application downloads	
New sounds for wireless charging and low battery	
New Gallery application animation allows faster loading	
USB debug whitelist	
Bug fixes and performance enhancements	
Android 4.3 (API level 18) (βλ. wikipedia/AndroidVersions)	
Bluetooth low energy support	
Bluetooth Audio/Video Remote Control Profile (AVRCP) 1.3 support	
OpenGL ES 3.0 support, allowing for improved game graphics	
Restricted access mode for new user profiles	
Filesystem write performance improvement by running fstrim command while device is idle	
Improvements to Photo Sphere	
Dial pad auto-complete in the Phone application	
Reworked camera UI, previously introduced on Google Play edition phones	
Addition of "App Ops", a fine-grained application permissions control system (hidden by default)	
4K resolution support	
Many security enhancements, performance enhancements, and bug fixes	
System-level support for geofencing and Wi-Fi scanning APIs	
Background Wi-Fi location still runs even when Wi-Fi is turned off	
Developer logging and analyzing enhancements	
Added support for five more languages	
Improved digital rights management (DRM) APIs	
Right-to-left (RTL) languages now supported	
Clock in the status bar disappears if clock is selected as lockscreen widget	

Android 4.3.1(API level 18) (βλ. wikipedia/AndroidVersions)

Bug fixes and small tweaks for the Nexus 7 LTE

2.1.15 ANDROID 4.4 KITKAT

Η επόμενη βασική έκδοση του *Android* ήταν η 4.1 με κωδικό όνομα «*KitKat*». Εκδόθηκε στις 31 Οκτωβρίου και την ακολούθησαν οι συμπληρωματικές εκδόσεις 4.4.1 στις 5 Δεκεμβρίου 2013 και η 4.4.2 στις 9 Δεκεμβρίου 2013. Η έκδοση αυτή προσαρμόστηκε ώστε να λειτουργεί σε μεγαλύτερο εύρος συσκευών από τις προηγούμενες εκδόσεις καθώς προτείνει σαν *minimum απαίτηση Ram* ίση με 512MB . Τα χαρακτηριστικά των εκδόσεων ήταν τα ακόλουθα:

Android 4.4 (API level 19) (βλ. wikipedia/AndroidVersions)

Refreshed interface with white elements instead of blue

Clock no longer shows bold hours, all digits are thin. The H, M, and S markings for the stopwatch and timer have been removed, leaving just the numbers.

Ability for applications to trigger translucency in the navigation and status bars

Ability for applications to use "immersive mode" to keep the navigation and status bars hidden while maintaining user interaction

Optimizations for performance on devices with lower specifications, including zRAM support and "low RAM" device API

NFC host card emulation, enabling a device to replace smart cards

Wireless printing capability

WebViews now based on Chromium engine (feature parity with Chrome for Android 30)

Removal of unofficial Flash Player support

Expanded functionality for notification listener services

Public API for developing and managing text messaging clients

New framework for UI transitions

Storage access framework for retrieving content and documents from other sources



Sensor batching, step detector and counter APIs

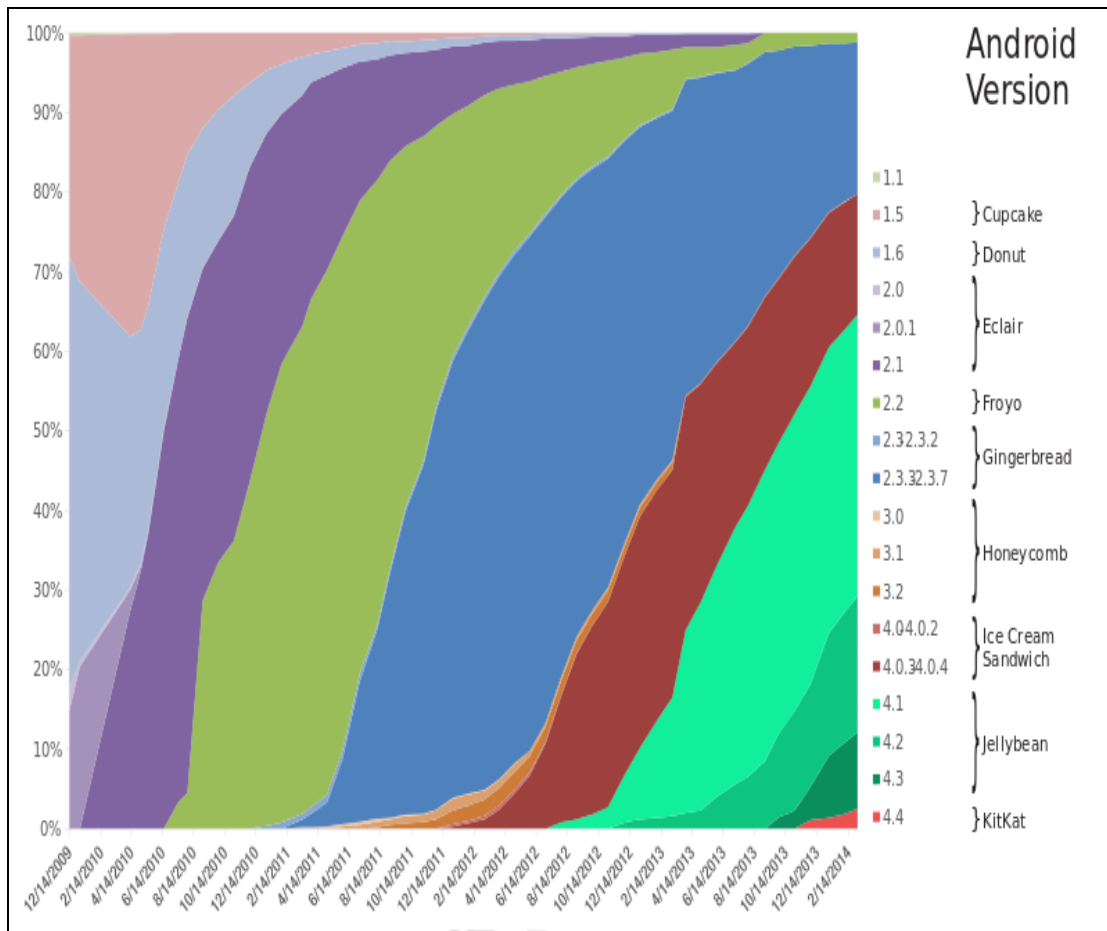
Settings application now makes it possible to select default text messaging and home (launcher) application

Audio tunneling, audio monitoring and loudness enhancer

Built-in screen recording feature (primarily for developers, as usage of ADB is required)
Native infrared blaster API
Expanded accessibility APIs and system-level closed captioning settings
New experimental runtime virtual machine, ART (not enabled by default)
Bluetooth Message Access Profile (MAP) support
Disabled access to battery statistics by third-party applications
Settings application no longer uses a multi-pane layout on devices with larger screens
Applications' write access to secondary storage (memory cards on devices with internal primary storage) is made possible but restricted to their designated private directories only, while full access to internal primary storage is still allowed through a separate application-level permission
Wi-Fi and mobile data activity (TX/RX) indicators are moved to quick settings
Android 4.4.1 (API level 19) (βλ. wikipedia/AndroidVersions)
Improvements to auto focus, white balance and HDR+ for the Nexus 5 camera
Better application compatibility for the experimental ART runtime
Camera application now loads Google+ Photos instead of Gallery when swiping away from the camera view
Miscellaneous improvements and bug fixes
Android 4.4.2 (API level 19) (βλ. wikipedia/AndroidVersions)
Further security enhancements and bug fixes
Removal of the "App Ops" application permissions control system, introduced in Android 4.3

2.1.16 ΣΥΣΧΕΤΙΣΜΟΣ ANDROID ΕΚΔΟΣΕΩΝ - ΣΥΣΚΕΥΩΝ

Στο παρακάτω γράφημα παρουσιάζονται οι εκδόσεις του *Android* και το ποσοστό των συσκευών που τις έφεραν κάθε στιγμή. Το γράφημα καταγράφει το ποσοστό αυτό από το τέλος του 2009 μέχρι τις αρχές του 2013.



Εικόνα 6: Συσχετισμός Android Συσκευών - Εκδόσεων

ΚΕΦΑΛΑΙΟ 3^ο

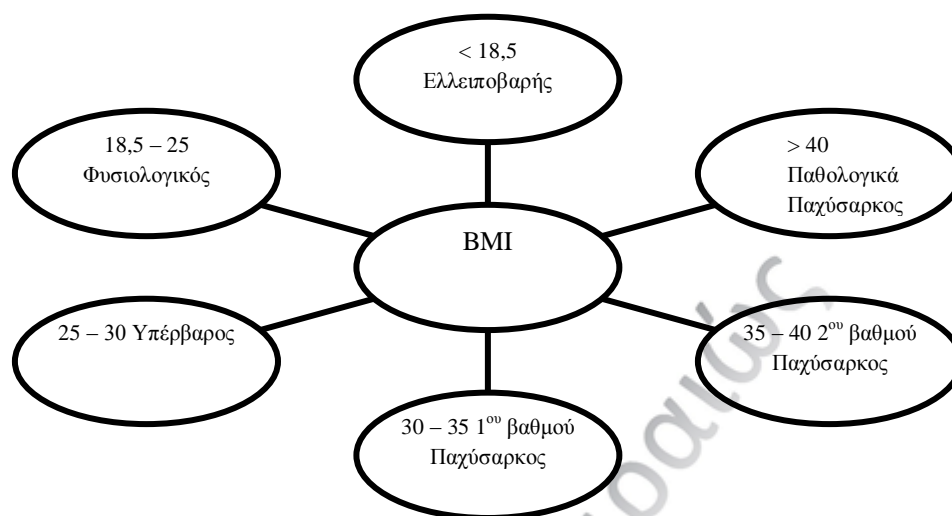
3.1 ΔΙΑΙΤΑ

Η εφαρμογή μας πραγματεύεται την εύρεση διαίτας από τον χρήστη έπειτα από την καταχώρηση προσωπικών του στοιχείων. Για να εξασφαλιστεί η εγκυρότητα της εφαρμογής απευθυνθήκαμε σε διαιτολόγο, που μας κατατόπισε πως μπορούμε να αξιολογήσουμε τα δεδομένα που δίνει ο χρήστης και να προκύψει από αυτά ένα εβδομαδιαίο διατροφικό πλάνο. Ο διαιτολόγος μας έδωσε τους παρακάτω τύπους:

$$\text{ΔΕΙΚΤΗΣ ΜΑΖΑΣ ΣΩΜΑΤΟΣ} = \text{βάρος (kg)} / \text{ύψος}^2 \text{ (m}^2\text{)} \quad (\text{Διεθνές σύστημα μονάδων (SI)})$$

Ο Δείκτης μάζας σώματος (ΔΜΣ, body mass index (BMI), ή Quetelet index) είναι μία γενική ιατρική ένδειξη για τον υπολογισμό του βαθμού παχυσαρκίας ενός ατόμου. Λόγω του εύκολου υπολογισμού του είναι ένα ευρέως διαδεδομένο διαγνωστικό εργαλείο των πιθανών προβλημάτων υγείας ενός ατόμου σε σχέση με το βάρος του. Δημιουργήθηκε το 1832[1] από τον Adolphe Quetelet. (βλ. wikipedia/BMI)

Ανάλογα με το αποτέλεσμα του *BMI* ο χρήστης κατηγοριοποιείται σε:



ΕΞΙΣΩΣΕΙΣ ΒΑΣΙΚΟΥ ΜΕΤΑΒΟΛΙΣΜΟΥ

a) Για άνδρες χρήστες : $66 + (13,7 * B) + (5 * Y) - (6,8 * H) = \dots \text{ kcal}$

b) Για γυναίκες χρήστες : $655 + (9,6 * B) + (1,8 * Y) - (4,7 * H) = \dots \text{ kcal}$

Όπου B = βάρος σε κιλά , Y = ύψος σε cm και H = ηλικία σε χρόνια

Basal metabolic rate (BMR), and the closely related resting metabolic rate (RMR), is the amount of energy expended daily by humans and other animals at rest. Rest is defined as existing in a neutrally temperate environment while in the post-absorptive state. In plants, different considerations apply. (βλ. [wikipedia/BMR](https://en.wikipedia.org/wiki/Basal_metabolic_rate))

Πριν χρησιμοποιήσουμε τους παραπάνω τύπους , ανάλογα με την κατηγορία που ανήκει ο χρήστης (σύμφωνα με το BMI) , κάνουμε την κατάλληλη προσαρμογή για το βάρος του :

$$\underline{\text{ΙΔΑΝΙΚΟ ΒΑΡΟΣ}} = 22,7 * Y^2$$

$\underline{\text{ΠΡΟΣΑΡΜΟΣΜΕΝΟ ΒΑΡΟΣ}} = (\text{Πραγματικό Βάρος} - \text{Ιδανικό Βάρος}) * (0,25) + \text{Ιδανικό Βάρος}$

- Παχύσαρκος (1ου βαθμού Υπέρβαρος , 2ου βαθμού Υπέρβαρος , Παθολογικά Υπέρβαρος) : Για το υπολογισμό του BMR χρησιμοποιείται το προσαρμοσμένο βάρος
- Ελλιποβαρής (Ελλιποβαρής) : Για το υπολογισμό του BMR χρησιμοποιείται το ιδανικό βάρος
- Φυσιολογικός (Φυσιολογικός , Υπέρβαρος) : Για το υπολογισμό του BMR χρησιμοποιείται το πραγματικό βάρος

Πέρα από το βάρος , το ύψος , το φύλλο και την ηλικία ζητάμε από τον χρήστη να μας προσδιορίσει την φυσική του δραστηριότητα εντός και εκτός του χώρου εργασίας του . Μπορεί να

επιλέξει μια από αυτές τις κατηγορίες (Ελαφριά , Μέτρια , Έντονη) και προκύπτει ένας συντελεστής που μας βοηθά να υπολογίσουμε τις ημερήσιες θερμιδικές ανάγκες του χρήστη :

ΦΥΣΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΕΚΤΟΣ ΕΡΓΑΣΙΑΣ	ΦΥΣΙΚΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑ ΣΤΗΝ ΕΡΓΑΣΙΑ					
	ΕΛΑΦΡΙΑ		ΜΕΤΡΙΑ		ΕΝΤΟΝΗ	
	ΑΝΔΡΕΣ	ΓΥΝΑΙΚΕΣ	ΑΝΔΡΕΣ	ΓΥΝΑΙΚΕΣ	ΑΝΔΡΕΣ	ΓΥΝΑΙΚΕΣ
ΕΛΑΦΡΙΑ	0,4	0,4	0,6	0,5	0,7	0,5
ΜΕΤΡΙΑ	0,5	0,5	0,7	0,6	0,8	0,6
ΕΝΤΟΝΗ	0,6	0,6	0,8	0,7	0,9	0,7

$$\text{ΗΜΕΡΗΣΙΕΣ ΘΕΡΜΙΔΙΚΕΣ ΑΝΑΓΚΕΣ} = \text{BMR} + (\text{BMR} * \text{Φυσική Δραστηριότητα}) = \dots \text{ kcal}$$

Ημερήσιες θερμιδικές ανάγκες είναι ουσιαστικά οι θερμίδες με τις οποίες φτιάχνουμε το διαιτολόγιο μας.

Έπειτα έχουμε ορίσει ότι το 30% των ημερήσιων θερμίδων θα καταναλώνεται από το πρωινό , το 50% από το μεσημεριανό και το 20% από το βραδινό . Μετά , βρίσκουμε ένα νέο νούμερο , που είναι οι ημερήσιες θερμίδες μειωμένες κατά 10% (βάλαμε 10% ενδεικτικά για να γίνει ομαλότερα και πιο σταδιακά η μείωση βάρους). Αυτό είναι το νούμερο δηλαδή με βάση το οποίο θα κινηθούμε και θα γίνει η εύρεση των τροφών. Οι προσαρμοσμένες πλέον ημερήσιες θερμίδες αποτελούν το ανώτερο όριο που δεν θα πρέπει να ξεπερνά το άθροισμα των θερμίδων του πρωινού , μεσημεριανού και βραδινού. Πρακτικά , αφού οι ημερήσιες θερμίδες μοιράζονται σε τρία γεύματα και έχουμε τις θερμίδες που έχουν αυτά , προσαρμόζουμε τις επιτρεπτές θερμίδες στα γεύματα αυτά και έχουμε το ίδιο αποτέλεσμα. (δηλαδή ημερήσιες θερμίδες - (10% * ημερήσιες θερμίδες) = (10% * (ημερήσιες θερμίδες *30%) [πρωινό]) + (10% * (ημερήσιες θερμίδες *50%) [μεσημεριανό]) + (10% * (ημερήσιες θερμίδες *20%) [βραδινό])

Παρακάτω ακολουθούν οι πίνακες με τις τροφές που χρησιμοποιούνται στην εφαρμογή μας και τις αντίστοιχες θερμίδες τους.

<u>ΠΡΩΙΝΑ</u>		
<u>id</u>	<u>ΟΝΟΜΑΤΑ</u>	<u>ΘΕΡΜΙΔΕΣ</u>
1	Τοστ με αυγό	255
2	Τοστ ζαμπόντυρί	400
3	Τοστ με 1 φέτα ζαμπόν	270
4	Ρυζόγαλο + Καφές αμερικανικός ή γαλλικός σκέτος (1 φλ.)	302
5	Τοστ με μπιφτέκι	440
6	Τοστ με 1 λεπτή φέτα χοιρομέρι	435
7	Αυγά Ομελέτα και τυρί	470
8	Αχλαδόμηλο μαρμελάδα (100γρ)	300
9	Γκούντα τυρί (100γρ) + γαλοπούλα βραστή στήθος (275γρ)	625
10	Δημητριακά με μέλι και ξηρούς καρπούς (100γρ.)	390
11	Δημητριακά με σταφίδες και χουρμάδες (100γρ.)	455
12	Δημητριακά Musli (100γρ.)	390
13	Δημητριακά με πίτουρο (1 φλ.)	210
14	Ενταμ τυρί (100 γρ.) + Ψωμάκι σκληρό για σάντουιτς 50γρ.	496
15	Γάλα συμπυκνωμένο ζαχαρούχο (1/2 φλιτζ. τσαγιού)	540
16	Γαλατόπιτα	300
17	Γάλα με φρούτα (μilkσέικ 1 ποτήρι)	190
18	Γάλα κατσικίσιο (1 φλιτζ. τσαγιού) + Φρυγανιές απλές μεγάλες (3)	454
19	Γάλα αγελάδος εβαπορέ (1/2 φλ. τσαγιού)	190
20	Εμενταλ τυρί (100 γρ.) + Ζαμπόν ημίπαχο (100 γρ.) + Ψωμάκι με σταφίδες 50γρ. + Λεμόνι χυμός (μισό φλιτζάνι)	883
21	Γραβιέρα τυρί (100 γρ.) + Ψωμάκι σκληρό για σάντουιτς 50γρ.	596
22	Καφές με ένα κ. γλυκού ζάχαρη + Φρυγανιές απλές μεγάλες (2)	198
23	Κέικ με σοκολάτα (1 κομμάτι 50γρ.)	325
24	Κέικ φρούτων (1 κομμάτι 50γρ.)	300
25	Κεφαλοτύρι (100 γρ.) + Ψωμάκι σκληρό για σάντουιτς 50γρ.	556
26	Κακάο ρόφημα (2+2 κουτ. κακάο/ζάχαρη)	205
27	Κουάκερ κρέμα (1 μπολ)	209
28	Κουφέτα με αμύγδαλα (100γρ.)	458
29	Κρουασάν με σοκολάτα (1 μεγάλο 75γρ.)	300
30	Κωκ (1 μεγάλο)	360
31	Λεμονάδα (2 λεμόνια με 2 κ. γλ. ζάχαρη)	100
32	Λεμόνι χυμός (μισό φλιτζάνι)	33
33	Λουκουμάδες Καφές αμερικανικός ή γαλλικός σκέτος (1 φλ.)	312
34	Μαρμελάδα (1 κουτ. γλυκού)	55

35	Μπέικον καπνιστό (100 γρ.)	770
36	Μπέικον τηγανητό (3 φέτες)	360
37	Μπέικον αλατισμένο (100 γρ.) + Ψωμάκι με σταφίδες 50γρ.	950
38	Μήλα στρούντελ με αμύγδαλα/καρύδια	460
39	Μηλόπιτα	395
40	Μπουγάτσα	360
41	Ντόνατς (1)	180
42	Πιτάκια με μαρμελάδα (1 μικρό)	200
43	Τσουρέκι μπριός (1 ατομικό)	230
44	Τυρί κότατζ (170 γρ.)	190
45	Τυρόπιτα	37
46	Φιστικοβούτυρο (100 γρ.)	660
47	Χάμπουργκερ (1 μεγάλο)	440
48	Χάμπουργκερ (1 μικρό)	275
49	Χαλβάς εμπορίου με βανίλια (100 γρ.)	506
50	Χορτόπιτα (1 μέτριο κομμάτι)	375

<u>ΜΕΣΗΜΕΡΙΑΝΑ</u>		
<u>id</u>	<u>ΟΝΟΜΑΤΑ</u>	<u>ΘΕΡΜΙΔΕΣ</u>
1	Αμπελοφάσουλα σαλάτα	340
2	Αρνί με αρακά	800
3	Αρνί με κολοκυθάκια	740
4	Αρνί μπριζόλες πανέ	700
5	Αρνί παϊδάκια σχάρας	370
6	Αρνί παϊδάκια τηγανητά	610
7	Αρνί φρικασέ + Ανθότυρο (100 γρ.) + Ελιές μαύρες (200 γρ.) + Ψωμί άσπρο (2 λεπτές φέτες)	1530
8	Ξιφίας ψητός	285
9	Γιουβέτσι με αρνί + Γκαζόζα (1 ποτήρι)	985
10	Γκούλας με κρέας + Γκούντα τυρί (μερίδα)	1110
11	Χυλοπίτες με τυρί και ντομάτα	570
12	Χταπόδι με σπανάκι	460
13	Χοιρινό με σέλινο	750
14	Χοιρινό σούβλας στήθος	435
15	Χοιρινό μπριζόλα σχάρας άπαχη	350
16	Χάμπουργκερ (1 μεγάλο)	440
17	Χυλοπίτες με κοτόπουλο	580
18	Θαλασσινά σε Κοκίγ με κρέμα γάλακτος και τυρί (1 μεγάλο όστρακο)	310
19	Ζυμαρικά απλά με κιμά	740

20	Ζυμαρικά απλά με σάλτσα μανιτάρια	680
21	Ζυμαρικά με βούτυρο & τυρί	535
22	Βοδινό φιλέτο σχάρας	300
23	Βοδινό ψητό φούρνου	440
24	Γαρίδες γιουβέτσι	540
25	Γαρίδες τηγανητές	330
26	Γαλοπούλα γεμιστή	495
27	Γκούλας με κρέας + Ψωμί άσπρο (2 λεπτές φέτες)	1090
28	Εσκαλόπ με ζαμπόν & πατάτες τηγανητές	650
29	Θαλασσινά σε βολοβάν (1 μεγάλο όστρακο)	395
30	Κόκορας κρασάτος (μερίδα) + Κόκα Κόλα (1 ποτήρι)	490
31	Κεφτέδες (4 κανονικοί)	460
32	Κανελόνια με κιμά γκρατινέ (4 μεγάλα)	1700
33	Καρμπονάρα σπαγγέτι (μερίδα 250γρ.)	840
34	Κότα ψητή σούβλας (1 μερίδα)	250
35	Κοτόσουπα με ρύζι αυγολέμονο (1 μερίδα 350γρ.)	290
36	Κουκιά φρέσκα λαδερά	430
37	Κουνέλι κοκκινιστό (μερίδα 260γρ.)	800
38	Κριθαράκι με σάλτσα ντομάτας (μερίδα 220γρ.)	520
39	Κουνουπίδι γιαχνί (μερίδα 260γρ.)	190
40	Λουκάνικα μοσχάρισμα ψητά (8 πιρουνιές)	528
41	Λυθρίνι ψητό χωρίς λάδι (1 κιλό)	1000
42	Μοσχάρι κοκκινιστό με πατάτες	710
43	Μοσχάρι σνίτσελ με πουρέ	600
44	Μοσχάρι φιλέτο σχάρας	250
45	Μοσχάρι φιλέτο με πατάτες	450
46	Μοσχάρι βραστό	255
47	Μπακαλιάρος τηγανητός με 1 κουταλιά σούπας σκορδαλιά	520
48	Μπιφτέκια ατμού χωρίς ψωμί (1 μεγάλο)	320
49	Ντομάτες γεμιστές με κιμά (1)	315
50	Ντολμάδες με λάχανο αβγολέμονο	400

<u>ΒΡΑΔΥΝΑ</u>		
<u>id</u>	<u>ΟΝΟΜΑΤΑ</u>	<u>ΘΕΡΜΙΔΕΣ</u>
1	Αχλάδι κομπόστα (1 μέτριο)	165
2	Αχλάδι τάρτα	385
3	Αυγά με ντομάτες (3 κ. σούπας)	165
4	Βερίκοκα κομπόστα (2 κομμάτια σιρόπι)	135

5	Βερίκοκα τάρτα	360
6	Βατόμουρα φρέσκα (ωμά 1 κιλό)	600
7	Βατόμουρα τάρτα	350
8	Γαλατόπιτα	300
9	Γκρέιπφρουτ (1 κομμάτι)	55
10	Γιαρμάδες (1 κιλό)	440
11	Γραβιέρα τυρί (μερίδα)	240
12	Δαμάσκηνα κομπόστα (5 κομμάτια)	190
13	Δαμάσκηνα ωμά (1 κιλό)	500
14	Ελιές πράσινες μέτριες (100 γρ.)	105
15	Εκλέρ σοκολάτας (1 μικρό)	200
16	Ζελέ χαμηλών θερμίδων (1 μπολ 220γρ.)	9
17	Ζαμπόν διαίτης (100 γρ.)	165
18	Ηλιόσποροι (ακαθάριστοι 100 γρ.)	302
19	Κεράσια (1 κιλό)	630
20	Κεράσια τάρτα με κρέμα (μερίδα 60γρ.)	180
21	Κολοκυθάκια βραστά σαλάτα (μερίδα 200γρ.)	180
22	Κάστανα βραστά ή ψητά (2)	110
23	Κάσιους (100γρ.)	600
24	Κασέρι τυρί (100 γρ.)	390
25	Κορόμηλα κομπόστα (3)	135
26	Κορν φλέικς (1 φλ. τσαγιού) + Γάλα αγελάδος εβαπορέ (1/2 φλ. τσαγιού)	300
27	Λέμονπái	400
28	Λουκουμάδες	310
29	Μανταρίνια (5)	225
30	Μούσλι (50 γρ.) + Γάλα αγελάδος εβαπορέ (1/2 φλ. τσαγιού)	405
31	Μουστοκούλουρο (1)	140
32	Μπισκότα σοκολάτας (1 διπλό)	165
33	Μπισκότα ginger (3 μικρά)	55
34	Μπισκότα σοκολάτας cookies με φιστίκια (2)	220
35	Μπισκότα απλά (1)	50
36	Μπρόκολο βραστό σαλάτα	230
37	Μπουγάτσα	360
38	Μήλα (1 φιρίκι)	65
39	Προφιτερόλ (1 μπολ)	540
40	Πορτοκάλι (1)	77
41	Ποπκορν (1 φλιτζάνι)	135
42	Παστέλι (100 γρ.)	640

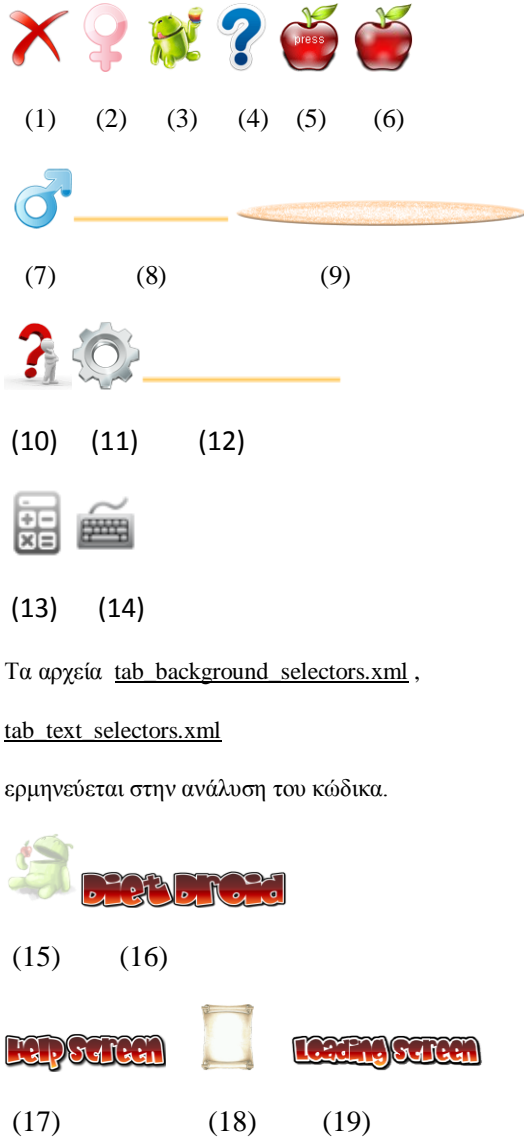
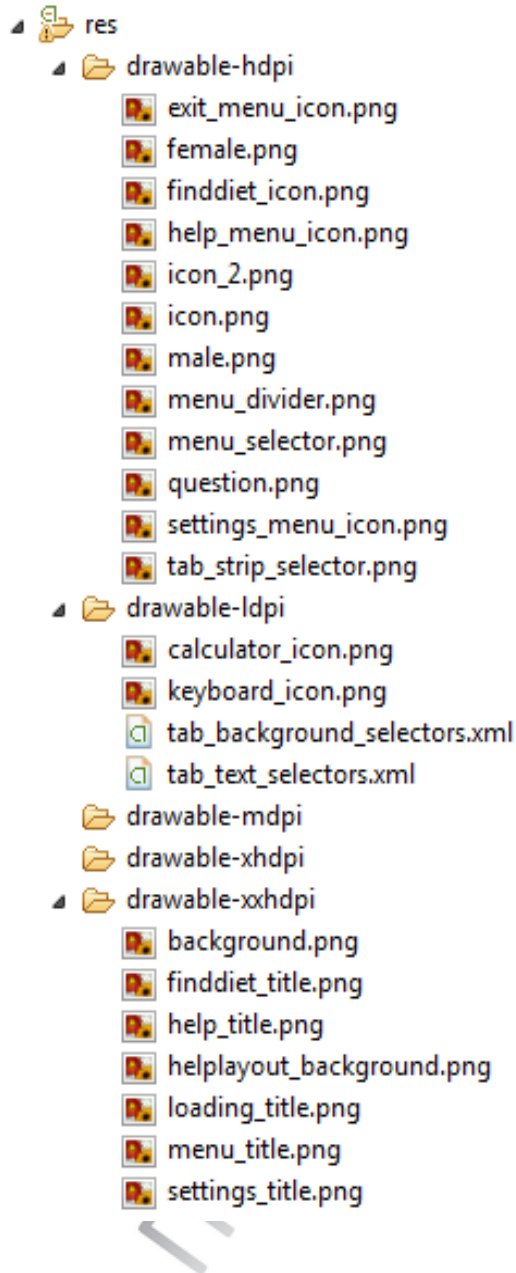
43	Πίτσα σπέσιαλ (1 μεγάλο τρίγωνο)	400
44	Ραβανί	365
45	Ρυζόγαλο (1 μπολ)	325
46	Ροδάκινα τάρτα	395
47	Σοκολατάκια (πολύ μικρά τα 5)	110
48	Σοκολατίνα πάστα	610
49	Σοκολατάκια τρούφες (1)	60
50	Σύκα ξερά (1)	65

ΚΕΦΑΛΑΙΟ 4^ο

4.1 ΔΟΜΗ DIETDROID

Η εφαρμογή μας έχει γραφτεί σε *Java* , *Xml* και σε ένα σημείο χρησιμοποιείται και *HTML* και *JavaScript*. Αποτελείται από 14 κλάσεις , από αυτές 7 σχετίζονται με τις *activities* της εφαρμογής και οι υπόλοιπες είναι συμπληρωματικές - βοηθητικές.. Επίσης αποτελείται από *xml* αρχεία , 4 για τον ορισμό τιμών των πόρων που χρησιμοποιεί το σύστημα (*colors* , *dimens* , *strings* , *styles*) , 1 για το αρχείο *Manifest* , 13 για να υλοποιούνται τα *layouts* των οθονών της εφαρμογής και 1 που αφορά το μενού της κεντρικής οθόνης. Ακόμα περιλαμβάνονται 2 αμεταγλώττιστα αρχεία *.gif* (*assets*) , 21 πόροι εικόνων , 2 ακόμη αρχεία *xml* που χρησιμοποιούνται για γραφιστικούς λόγους και 2 αρχεία *.txt* (*raw*) που μας βοηθούν στη παρουσίαση των *.gif* και στην *help screen* .

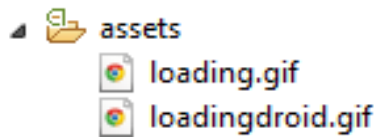
4.2 ΑΡΧΕΙΑ ΓΡΑΦΙΚΩΝ & ΑΡΧΕΙΑ ΠΟΡΩΝ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΣΧΕΔΙΑΣΗΣ



Τα αρχεία [tab_background_selectors.xml](#) ,
[tab_text_selectors.xml](#)

ερμηνεύεται στην ανάλυση του κώδικα.

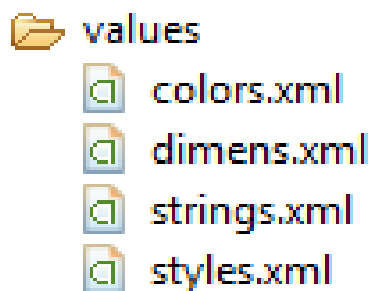
4.3 ΑΜΕΤΑΓΛΩΤΤΙΣΤΑ ΑΡΧΕΙΑ ΠΟΡΩΝ



(1)

(2)

4.4 ΔΙΑΦΟΡΟΙ ΤΥΠΟΙ ΠΟΡΩΝ (ΧΡΩΜΑΤΑ , ΣΥΜΒΟΛΟΣΕΙΡΕΣ ΚΕΙΜΕΝΟΥ , ΣΤΥΛ , ΔΙΑΣΤΑΣΕΙΣ)



Αυτός ο υποκατάλογος οργανώνει τους διαφορετικούς τύπους πόρων . Στην εφαρμογή μας περιέχονται πόροι χρωμάτων , διαστάσεων , συμβολοσειρών , στυλ μορφοποίησης αντίστοιχα.

4.4.1 Colors.xml

Το αρχείο « *colors.xml* » περιέχει τα ονόματα των χρωμάτων , την δεκαεξαδική τους σημειογραφία (ένας συνδυασμός κόκκινου - πράσινου – μπλε χρώματος) και σχόλια για τον προγραμματιστή για το πιο χρώμα απεικονίζεται . Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

4.4.2 Dimens.xml

Το αρχείο « *dimens.xml* » περιλαμβάνει τα ονόματα και τις διαστάσεις που χρησιμοποιούνται στην εφαρμογή . Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

4.4.3 Strings.xml

Το αρχείο « *strings.xml* » περιλαμβάνει όλες εκείνες τις συμβολοσειρές που χρησιμοποιούνται στο πρόγραμμα μας . Φαίνονται τα ονόματα των συμβολοσειρών , τα ονόματα των πινάκων συμβολοσειρών , οι ίδιες οι συμβολοσειρές και σε σχόλια , για δικής μας βοήθεια , φαίνεται που γίνεται η χρήση των συμβολοσειρών. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.


4.4.4 Styles.xml


Το αρχείο « *styles.xml* » περιλαμβάνει όλα τα διαφορετικά στυλ εμφάνισης που χρησιμοποιεί η εφαρμογή μας . Έχουμε προσθέσει αρκετά δικά μας στυλ , που χρησιμοποιούνται στα *alertdialogs* , στα *textviews* και *inputtexts* (π.χ. *themeForAlertDialog* , *textviews_settings* , *edittexts_settings*) της εφαρμογής μας.

Επίσης έχει δημιουργηθεί και ένα απλό στυλ για πού χρησιμοποιείται σε όλες τις *activities* της εφαρμογής προκειμένου να έχουν παρόμοια εμφάνιση (*themeForActivities* - η χρήση της γίνεται στο *AndroidManifest*). Σε κάθε στυλ που δημιουργήθηκε έχουν προστεθεί *xml attributes* με την ονομασία τους και με την τιμή του χαρακτηριστικού που επιθυμούμε. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

4.5 ΑΚΑΤΕΡΓΑΣΤΑ ΑΡΧΕΙΑ ΠΟΡΩΝ



 `format_loadinglayout.txt`

 `help_text.txt`

Αυτός ο υποκατάλογος οργανώνει τα ακατέργαστα αρχεία που

μπορεί να χρησιμοποιεί η εφαρμογή, π.χ. αρχεία ήχου, βίντεο.

Στη περίπτωση μας το (1) περιέχει κώδικα που θα

χρησιμοποιηθεί και το (2) περιέχει απλό κείμενο.

4.5.1 Format_loadinglayout.xml

Το αρχείο « *format_loadinglayout.txt* » είναι ένα αρχείο text που περιλαμβάνει κώδικα *HTML* και *JAVASCRIPT*. Η εφαρμογή μας καλεί το αρχείο αυτό, κατά την πρώτη δραστηριότητα της (*LoadingActivity*), όταν θέλουμε να παρουσιάσουμε στο χρήστη την «loading page». Χρησιμοποιήθηκαν οι γλώσσες αυτές επί σκοπό στο συγκεκριμένο σημείο καθώς, τα αρχεία «*loadingdroid.gif*» και «*loading.gif*» που βρίσκονται στο φάκελο *assets*, είναι αρχεία *.gif* και δεν μπορούν να παρουσιαστούν σαν απλές εικόνες *.png* ή *.jpg* σε συσκευή android. Γι αυτό χρησιμοποιήθηκε η επιλογή της δημιουργίας μια *WebView* στη πρώτη δραστηριότητα της εφαρμογής, καθώς αναπαριστά την λειτουργία ενός *browser* και οι *browsers* μπορούν να δείξουν τα αρχεία *.gif*. Τώρα στο αρχείο βλέπουμε ότι χρησιμοποιείται μια συνάρτηση η « *wh()* » για να πάρουμε το ύψος και το μήκος της οθόνης, τα οποία έπειτα ορίζονται σαν χαρακτηριστικά των *gifs* που έχουμε για *images*, ώστε να μπορούν τα τελευταία να απεικονιστούν στο κέντρο με την εντολή «*align='middle'*». Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

4.5.2 Help_text.txt

The application "DietDroid" belongs to Antonios Manettas of George, graduate student in the Department of Informatics at University of Piraeus. The application is part of the thesis of his master in Informatics. The application proposes to the user a diet, based on tables "food - calories". The calories should daily take the user resulting from his "BMI". The "BMI" has several parameters, such as weight, height, sex, etc. and the calculation is based on truthful data supplied by the user in the application.

Το αρχείο « *help_text.txt* » είναι ένα αρχείο text που περιλαμβάνει απλά σκέτο κείμενο που χρησιμοποιείται από την δραστηριότητα (*HelpActivity*) καθαρά για λόγους ενημέρωσης του χρήστη για την εφαρμογή.

ΚΕΦΑΛΑΙΟ 5^ο

5.1 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΒΟΗΘΗΤΙΚΩΝ ΚΛΑΣΕΩΝ

5.1.1 Βοηθητικές Κλάσεις

Παρακάτω ακολουθεί η παρουσίαση των βοηθητικών κλάσεων της εφαρμογής μας. Βοηθητικές κλάσεις είναι αυτές που καλούνται από τις κυρίες κλάσεις μας , για να πραγματοποιήσουν μια ενέργεια , μια δράση . Οι ενέργειες αυτές και κατά συνέπεια και ο κώδικας των βοηθητικών κλάσεων θα μπορούσε να έχει συγχωνευτεί με αυτόν των κύριων κλάσεων , αλλά είναι καλύτερη πρακτική ο διαμερισμός του κώδικα και η χρήση με κλήσεις στο σημείο που επιθυμούμε , ιδίως και αφού η χρήση του κώδικα των βοηθητικών κλάσεων επαναλαμβάνεται.

Οι βοηθητικές μας κλάσεις δεν έχουν κάποια μορφή / απεικόνιση στο *user interface* του κινητού . Είναι ενέργειες που συμπληρώνουν τα κύρια κομμάτια κώδικα. Κατά συνέπεια δεν σχετίζονται με κάποιο *layout* αρχείο , για αυτό και παρουσιάζονται μόνες τους στο κομμάτι αυτό.

5.1.2 ExitDroid.java

Η βοηθητική κλάση *ExitDroid* καλείται όταν επιδιώκεται από τον χρήστη η έξοδος από την εφαρμογή. Έξοδος μπορεί να γίνει κατά την *LoadingActivity* (οθόνη φόρτωσης πατώντας το *BackButton*) και στη *MenuActivity* (στο μενού πατώντας το κουμπί *Menu Button* του κινητού ή επιλέγοντας την έξοδο από τα στοιχεία του μενού) . Η κλάση αυτή δέχεται σαν όρισμα την *Activity* που την κάλεσε , κρατεί το περιεχόμενο της για να μπορέσει να εμφανιστεί πάνω σε αυτή και δημιουργεί ένα κατασκευαστή (*Builder*) για το παράθυρο διαλόγου (*AlertDialog*). Ορίζεται το *style* εμφάνισης του παραθύρου πάνω στη *Activity* και έπειτα ο τίτλος του παραθύρου , το μήνυμα και η εικόνα που θα εμφανίζει.

AlertDialog - Ένα παράθυρο διαλόγου με ένα , δυο ή τρεις μηχανισμούς ελέγχου *Button* . Αυτός ο τύπος διαλόγου χρησιμοποιείται συχνά για να πάρουμε επιβεβαίωση ή λεπτομέρειες από τον χρήστη για μια λειτουργία π.χ. έξοδο από την εφαρμογή

Εμείς έχουμε δυο μηχανισμούς ελέγχου *Buttons* στο *AlertDialog* μας (*Yes* , *No*). Ορίζουμε και στις δυο επιπτώσεις *Listeners* πάνω στα *Buttons* αυτά και με το *click* του χρήστη ενεργοποιείται ένας χειρίστης (*Handler*) για να εφαρμόσει μια μικρή καθυστέρηση στη εκτέλεση του προγράμματος (0,5 sec) ώστε να εμφανιστεί σε *Toast Message* η επιλογή που πάτησε ο χρήστης . Όταν ο χρήστης πατήσει *Yes* ο διάλογος κλείνει (*dialog.cancel*) , εμφανίζεται το *Toast Message (You clicked on YES)* και έπειτα ολοκληρώνεται η *Activity (finish())* και κλείνει και η εφαρμογή (*System.exit(0)*) . Όταν πατήσει *No* ο διάλογος κλείνει εμφανίζεται το *Toast Message (You clicked on No)* και συνεχίζεται η εφαρμογή από κει που σταμάτησε πριν την εμφάνιση του *AlertDialog*. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.



Εικόνα 7: Exit Dialog

MENU SCREEN

Find Diet

Settings

Help

Exit

You clicked on NO

Εικόνα 8: Exit Dialog (Answer No)

MENU SCREEN

Find Diet

Settings

Help

Exit

You clicked on YES

Εικόνα 9: Exit Dialog (Answer Yes)

5.1.3 FullScreen.java

Η βοηθητική κλάση *FullScreen* καλείται από όλες τις *Activity*. Δηλαδή οποιαδήποτε οθόνη εμφανίζεται στο χρήστη κατά την εκτέλεση της εφαρμογής περιέχει την κλάση αυτή (*LoadingActivity*, *MenuActivity*, *HelpActivity*, *SettingsActivity*, *FindDietActivity*, *ResultsActivity*). Ουσιαστικά η κλάση αυτή χρησιμοποιήθηκε για να καλύπτει η εφαρμογή ολόκληρη την οθόνη του κινητού. Με τη μέθοδο *requestWindowFeature* γίνεται επέκταση των στοιχείων του παραθύρου της εφαρμογής σε ολόκληρη την οθόνη και βάζοντας στον *windowsManager* παραμέτρους για κάλυψη ολόκληρης της οθόνης (*LayoutParams.FLAG_FULLSCREEN*) καταφέρνουμε να κρύψουμε και την μπάρα ειδοποιήσεων (*Notification Bar*, μπάρα εμφάνισης σήματος, μπαταρίας, ασύρματου δικτύου κ.α.). Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.4 InputStreamToString.java

Η βοηθητική κλάση *InputStreamToString* καλείται σε δυο περιπτώσεις μέσα στη εφαρμογή μας. Η πρώτη περίπτωση είναι κατά την *LoadingActivity*, ώστε να μας βοηθήσει να εισάγουμε ένα ακατέργαστο πόρο κειμένου (*raw*), μετατρέποντας τον σε αντικείμενο *String*, βοηθώντας την *WebView* μας σε εκείνο το σημείο να φορτώσει ένα αρχείο *html/text* και μας δείξει τα αρχεία *gif*. Η δεύτερη χρήση της είναι στη *HelpActivity*, που πάλι πρέπει να εισάγουμε ένα ακατέργαστο πόρο κειμένου (*raw*), μετατρέποντας τον σε αντικείμενο *String* και εμφανίζοντας το αυτή τη φορά σε ένα *TextView*, αποτελώντας το κείμενο βοήθειας της εφαρμογής μας. Τώρα όσον αφορά την κλάση *InputStreamToString* δημιουργούμε ένα αντικείμενο *Buffer* τύπου *String* και βάζουμε ένα άλλο αντικείμενο *DataInputStream* να διαβάσει γραμμή το *txt* αρχείο μας και μέχρι να τελειώσουν όλες οι γραμμές να το προσθέτει στο *Buffer* αντικείμενο μας. Έτσι έχουμε ουσιαστικά όλο το κείμενο μας αποθηκευμένο ανά πάσα στιγμή σε ένα αντικείμενο *Buffer* τύπου *String*, το οποίο μετατρέπουμε σε αντικείμενο *String*, στο κώδικα στα σημεία που καλούμε ένα αντικείμενο της κλάσης *InputStreamToString* και το χρησιμοποιούμε αναλόγως όπως προαναφέρθηκε. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.5 JumpToActivity.java

Η βοηθητική κλάση *JumpToActivity* καλείται όταν θέλουμε να μεταβούμε από μια δραστηριότητα σε μια άλλη. Για παράδειγμα όταν τελειώνει η *LoadingActivity* (οθόνη φόρτωσης) και μεταβαίνουμε στη *MenuActivity* (μενού). Στη πραγματικότητα χρησιμοποιείται μόνο στη συγκεκριμένη περίπτωση γιατί θέλουμε να γίνει η μετάβαση αυτή με κάποια καθυστέρηση. Στις άλλες περιπτώσεις, όπως π.χ. τη μετάβαση από το μενού στη οθόνη βοήθειας (*MenuActivity - HelpActivity*), χρησιμοποιείται η μέθοδος *startActivity(new Intent(Context packageContext, Class<?> cls))*. Η κλάση *JumpToActivity* έχει τρία ορίσματα, την *Activity* που την κάλεσε, την *Activity* στην οποία θα μεταφερθούμε και την καθυστέρηση που θα υπάρξει. Ένας χειριστής *Handler* είναι υπεύθυνος να εφαρμόσει την καθυστέρηση που είναι 13 sec, ώστε να φορτωθεί σωστά και σε πλήρη ανάπτυξη όλα τα *gif* αρχεία της *LoadingActivity* (οθόνη φόρτωσης). Έπειτα μεταβαίνουμε από την παλιά δραστηριότητα στη καινούργια και τερματίζουμε την παλιά ώστε με το *Back Button* να μην μπορούμε να γυρίσουμε σε αυτή (να αφαιρεθεί από τη στοιβά των δραστηριοτήτων). Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.6 Prwina.java

Η βοηθητική κλάση *Prwina* δημιουργείται για να χρησιμοποιηθεί από την κλάση *DatabaseHandler* στην υλοποίηση της βάσης δεδομένων της εφαρμογής μας. Ουσιαστικά στην κλάση αυτή είναι όλες οι *getter* και οι *setter* μέθοδοι κάθε μεμονωμένου πρωινού που έχουμε εγγράψει και θα ανακαλέσουμε ή που θα γράψουμε στη βάση μας.

Αρχικά ορίζουμε τρεις μεταβλητές που αντιπροσωπεύουν τα κελιά του αντίστοιχου πίνακα με τα πρωινά μας (*TABLE_PRWINA*). Έπειτα έχουμε τρεις *constructors* που μας βοηθούν να δημιουργούμε

αντικείμενα τύπου *Prwina* , ανάλογα με τα ορίσματα που τους βάζουμε . Ακολουθούν οι *getter* και οι *setter* μέθοδοι . Αρχικά για το *id* των πρωινών και έπειτα για το όνομα τους και τις θερμίδες που το καθένα έχει . Οι *setter* μέθοδοι θα χρησιμοποιηθούν μια φορά στην εφαρμογή μας όταν και δημιουργείται η στατική βάση δεδομένων μας (κατά την πρώτη εκτέλεση του προγράμματος) , ενώ οι *getter* μέθοδοι κάθε φορά που επιδιώκουμε να εξάγουμε αποτελέσματα από την βάση στην δραστηριότητα *ResultsActivity*. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.7 Mesimeriana.java

Η βοηθητική κλάση *Mesimeriana* δημιουργείται για να χρησιμοποιηθεί από την κλάση *DatabaseHandler* στην υλοποίηση της βάσης δεδομένων της εφαρμογής μας . Ουσιαστικά στην κλάση αυτή είναι όλες οι *getter* και οι *setter* μέθοδοι κάθε μεμονωμένου μεσημεριανού που έχουμε εγγράψει και θα ανακαλέσουμε ή που θα γράψουμε στη βάση μας .

Αρχικά ορίζουμε τρεις μεταβλητές που αντιπροσωπεύουν τα κελιά του αντίστοιχου πίνακα με τα μεσημεριανά μας (*TABLE_MESIMERIANA*). Έπειτα έχουμε τρεις *constructors* που μας βοηθούν να δημιουργούμε αντικείμενα τύπου *Mesimeriana*, ανάλογα με τα ορίσματα που τους βάζουμε . Ακολουθούν οι *getter* και οι *setter* μέθοδοι . Αρχικά για το *id* των μεσημεριανών και έπειτα για το όνομα τους και τις θερμίδες που το καθένα έχει . Οι *setter* μέθοδοι θα χρησιμοποιηθούν μια φορά στην εφαρμογή μας όταν και δημιουργείται η στατική βάση δεδομένων μας (κατά την πρώτη εκτέλεση του προγράμματος) , ενώ οι *getter* μέθοδοι κάθε φορά που επιδιώκουμε να εξάγουμε αποτελέσματα από την βάση στην δραστηριότητα *ResultsActivity*. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.8 Bradyna.java

Η βοηθητική κλάση *Bradyna* δημιουργείται για να χρησιμοποιηθεί από την κλάση *DatabaseHandler* στην υλοποίηση της βάσης δεδομένων της εφαρμογής μας . Ουσιαστικά στην κλάση αυτή είναι όλες οι *getter* και οι *setter* μέθοδοι κάθε μεμονωμένου βραδινού που έχουμε εγγράψει και θα ανακαλέσουμε ή που θα γράψουμε στη βάση μας .

Αρχικά ορίζουμε τρεις μεταβλητές που αντιπροσωπεύουν τα κελιά του αντίστοιχου πίνακα με τα βραδινά μας (*TABLE_BRADYNA*). Έπειτα έχουμε τρεις *constructors* που μας βοηθούν να δημιουργούμε αντικείμενα τύπου *Bradyna* , ανάλογα με τα ορίσματα που τους βάζουμε . Ακολουθούν οι *getter* και οι *setter* μέθοδοι . Αρχικά για το *id* των βραδινών και έπειτα για το όνομα τους και τις θερμίδες που το καθένα έχει . Οι *setter* μέθοδοι θα χρησιμοποιηθούν μια φορά στην εφαρμογή μας όταν και δημιουργείται η στατική βάση δεδομένων μας (κατά την πρώτη εκτέλεση του προγράμματος) , ενώ οι *getter* μέθοδοι κάθε φορά που επιδιώκουμε να εξάγουμε αποτελέσματα από την βάση στην δραστηριότητα *ResultsActivity*. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.1.9 DatabaseHandler.java

Η βοηθητική κλάση *DatabaseHandler* είναι μια πολύ σημαντική κλάση , καθώς ενσωματώνει όλες τις ενέργειες που σχετίζονται με τη βάση δεδομένων μας . Περιλαμβάνει ενέργειες όπως η δημιουργία της βάσης , η ανανέωση / αναβάθμιση της , η δημιουργία πινάκων , η εισαγωγή τιμών στα κελιά των πινάκων αυτών και εκτελείσει ερωτημάτων στη βάση ,ώστε να εξαχθούν αποτελέσματα . Η κλάση αυτή στην εφαρμογή μας καλείται σε δυο περιπτώσεις. Πρώτον καλείται όταν δημιουργείται η βάση δεδομένων μας , κατά την πρώτη φορά εκτέλεσης της εφαρμογής , από την *MenuActivity* . Δεύτερον καλείται από την *ResultsActivity* , όταν έχουμε καταχωρήσει τα δεδομένα μας στην εφαρμογή και με βάση αυτά (και κάποιους υπολογισμούς) γίνονται ερωτήσεις (*queries*) στη βάση δεδομένων για να εξαχθούν αποτελέσματα .

Η βοηθητική μας κλάση αποτελεί ουσιαστικά ένα χειριστή βάσης δεδομένων , που επεκτείνει την κλάση *SQLiteOpenHelper* (υπεύθυνη για την δημιουργία και διαχείριση της βάσης μας και των

εκδόσεων της). Έπειτα προστίθενται κάποιες μεταβλητές, όπως η έκδοση της βάσης, το όνομα της, τα ονόματα των πινάκων και των κελιών τους. Ακολούθως υλοποιούνται δυο μέθοδοι, αναγκαστικές εξαιτίας της κλάσης *SQLiteOpenHelper* που αφορούν την δημιουργία της βάσης και την αναβάθμιση της. Στην *onCreate(SQLiteDatabase db)* ουσιαστικά γράφεται τρία *queries* στη βάση μας και εκτελούνται. Κάθε ένα από αυτά αφορά την δημιουργία ενός πίνακα (αν δεν υπάρχει) ορίζοντας όνομα πίνακα, ονόματα και τύπο κελιών. Μετά τα *queries* εκτελούνται από την βάση μας και μας δίνουν τρεις πίνακες, πρωινά, μεσημεριανά, βραδινά. Στη μέθοδο *onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)* η βάση μας εκτελεί τρία *queries* πάλι που καταστρέφει τους πίνακες μας και καλεί τη μέθοδο *onCreate(SQLiteDatabase db)* για να τους ξαναδημιουργήσει με τη νεότερη τους έκδοση. Στην εφαρμογή μας πρακτικά η μέθοδος αυτή δεν καλείται καθώς έχουμε δημιουργήσει μια φορά μια βάση δεδομένων στατική, που απλά εκτελούνται σε αυτή *queries*, άρα δεν γίνεται και κάποια προσθαφαίρεση στοιχείων μέσα από το πρόγραμμα στη συνέχεια και δεν αναβαθμίζεται η βάση μας.

Η κύρια λειτουργία της βάσης μας εστιάζεται στη δημιουργία της (μια φορά), στη πρόσθεση στοιχείων (μια φορά) στους πίνακες και στη εκτέλεση ερωτημάτων στους πίνακες αυτούς (πολλές φορές). Η διαδικασία πρόσθεσης στοιχείων είναι κοινή και για τους τρεις πίνακες μας. Χρησιμοποιείται μια μέθοδος *addPrwina(Prwina prwina)* ή *addMesimeriana(Mesimeriana mesimeriana)* ή *addBradyna(Bradyna bradyna)* αντίστοιχα. Αυτές καλούν άλλες μεθόδους *getWritableDatabase()* (για να επιτρέψουν το «διάβασμα» και την «εγγραφή» στοιχείων από τη βάση μας), δημιουργούν αντικείμενα «πακέτων» τιμών με τις *getter* μεθόδους των αντικειμένων που έχουμε βάλει σαν ορίσματα (ανάλογα με τον εκάστοτε πίνακα) και τα εγγράφουν στους αντιστοιχούν πίνακες και τέλος κλείνουν τη βάση μας.

Η λήψη στοιχείων από την βάση γίνεται μόνο στην κλάση *ResultsActivity*. Επιθυμούμε να πάρουμε αποτελέσματα σύμφωνα με τα στοιχεία που έχει δηλώσει ο χρήστης στη κλάση *FindDietActivity*. Η διαδικασία είναι πάλι κοινή και για τους τρεις πίνακες μας. Το μόνο που διαφέρει είναι τα ποσοστά που χρησιμοποιούμε στα *queries*, που προκύπτουν από τα όσα μας έχει υποδείξει ο διατολόγος για τον υπολογισμό της δίαιτας. Για να λάβουμε τα στοιχεία από τη βάση ως απάντηση από τα *queries*, χρησιμοποιούμε λίστες. Έτσι οι μέθοδοι μας είναι τύπου *List<Prwina>*, *List<Mesimeriana>* και *List<Bradyna>* και τα ονόματα των μεθόδων είναι *getPrwina()*, *getMesimeriana()* και *getBradyna()* αντίστοιχα. Εκτελούμε ένα *query*, έχοντας ανοίξει τη βάση μας, που λέει στη βάση από το αντίστοιχο κάθε φορά πίνακα να πάρει όλα τα πρωινά, μεσημεριανά ή βραδινά που να είναι μικρότερα από το αποτέλεσμα του τύπου που μας έδωσε ο διατολόγος (που σχετίζεται με γεύματα, ποσοστό σταδιακής δίαιτας και ημερήσιες θερμίδες). Μετά μας επιστρέφεται ένα αντικείμενο τύπου *Cursor* που δείχνει στα αποτελέσματα του *query* που έχουν τη μορφή στη δική μας περίπτωση του συγκεκριμένου πίνακα που εκτελέστηκε το *query*, δηλαδή πίνακας με κελί 1, κελί 2, κελί 3. Έπειτα μετατρέπουμε όλα τα κελιά του πίνακα αποτελεσμάτων σε *String* και εισάγουμε στη λίστα μας το αντικείμενο που αφορά το γεύμα μας κάθε φορά (με τις 3 κάθε φορά επιπλέον πληροφορίες του *id,name,thermides*). Αυτό εφαρμόζεται και στα πρωινά και στα μεσημεριανά και στα βραδινά γεύματα, προκύπτουν αποτελέσματα σε λίστες με όλες τις πληροφορίες που χρειαζόμαστε και στη συνέχεια στις λίστες αυτές θα εφαρμοστεί η *random()* ώστε να μας εξάγει τυχαία κάθε φορά αποτελέσματα. Ο κώδικας του αρχείου βρίσκεται στο παράρτημα της εργασίας.

5.2 ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΚΥΡΙΩΝ ΚΛΑΣΕΩΝ

Παρακατω ακολουθεί η ανάλυση του κώδικα των κύριων κλάσεων της εφαρμογής μας. Ο κώδικας αυτός είναι ουσιαστικός για την εφαρμογή μας, είναι η ραχοκοκαλιά της, σχετίζεται με το τι βλέπει και το τι κάνει ο χρήστης στην εφαρμογή, αποτυπώνει τις διάφορες οθόνες μέσω των *layout* που χρησιμοποιούνται από την καθεμιά κλάση (και από τον ίδιο τον κώδικα). Έτσι δεν μπορεί να παρουσιαστεί η κάθε κλάση ξεχωριστά από τα *layouts xml* αρχεία που σχετίζεται. Αξίζει να σημειωθεί ότι κάθε ξεχωριστή κύρια οθόνη της εφαρμογής μας σχετίζεται με μια κύρια κλάση / δραστηριότητα της εφαρμογής μας. Συνολικά οι κυρίες δραστηριότητες της εφαρμογής μας είναι 7.

Στο παρακάτω διάγραμμα φαίνεται η συνολική κίνηση μεταξύ δραστηριοτήτων της εφαρμογής, ένα σχεδιάγραμμα που αποτυπώνει τι μπορεί να κάνει ο χρήστης, που μπορεί να μεταφερθεί και με ποια

κουμπιά , ποιες είναι οι επιτρεπτές ενέργειες. Στη συνέχεια , με την ανάλυση του κώδικα των κύριων κλάσεων γίνεται αντιληπτό πως γίνονται οι μεταβάσεις που δείχνει το διάγραμμα.

Διάγραμμα των Οθονών - Activities

LoadingActivity = 1

MenuActivity = 2

SettingsActivity = 3

HelpActivity = 4

Exit (Dialog) = 5

ResultsActivity = 6

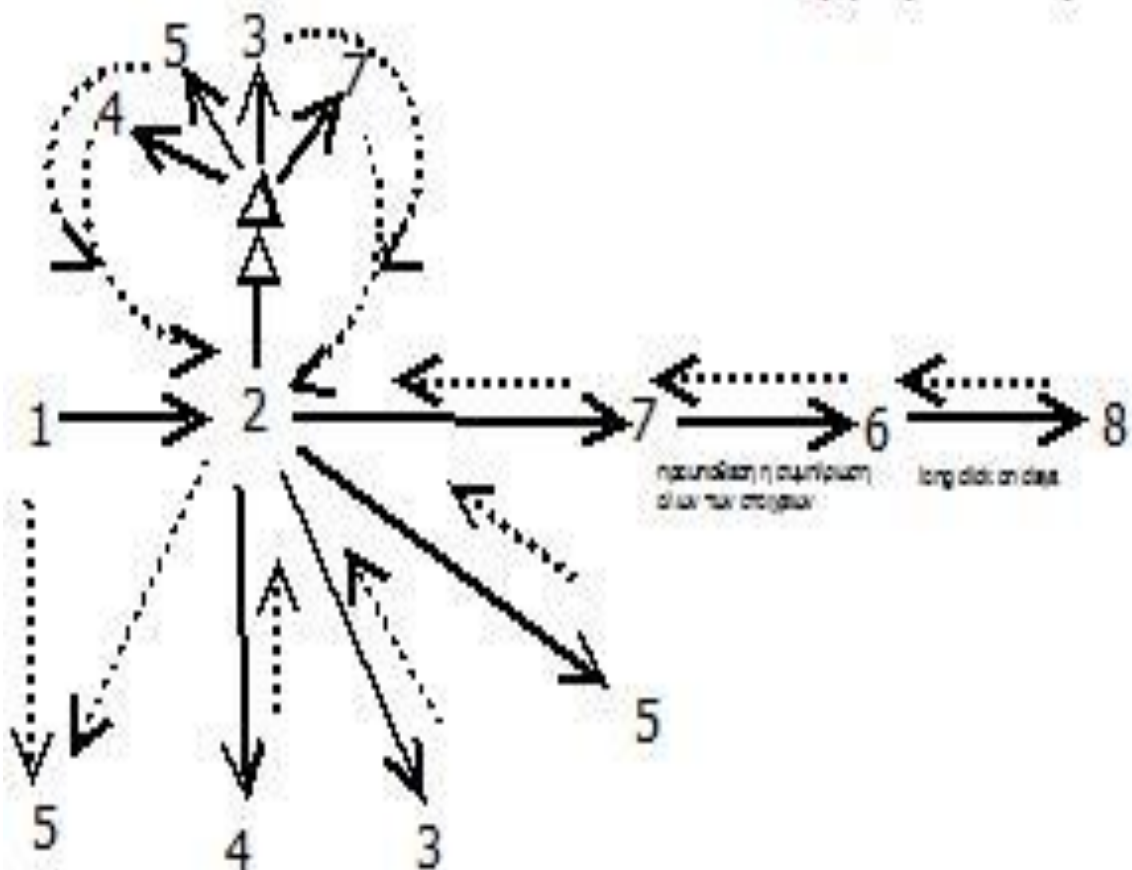
FindDietActivity = 7

back button =➤

forward move = ➡

menu button (mobile) = ➡➡

Calendar app (mobile) = 8



Εικόνα 10: Διάγραμμα οθονών – δραστηριοτήτων

5.2.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.dietdroid"
    android:versionCode="1"
    android:versionName="1.0" >
```

Ορίζουμε την έκδοση της εφαρμογής και το όνομα της έκδοσης της εφαρμογής.

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="19" />
```

Ορίζουμε ποια είναι η μικρότερη android έκδοση στην οποία μπορεί να τρέξει η εφαρμογή και με βάση ποια γίνεται το compile της .

```
<application
    android:allowBackup="true"
    android:icon="@drawable/icon"
    android:label="@string/application_name"
    android:theme="@style/themeForActivities" >
    <activity
        android:name="com.example.dietdroid.LoadingActivity"
        android:screenOrientation="portrait"
        android:stateNotNeeded="true"
        android:noHistory="true"
        android:launchMode="singleInstance">
```

Δεν κρατάμε ιστορικά ίχνη της εφαρμογής , η LoadingActivity τελειώνει μόνιμα . Δεν μας ενδιαφέρει να κρατήσουμε την προηγούμενη κατάσταση της , σβήνεται από τη στοίβα δραστηριοτήτων. Θέλουμε μόνο προσανατολισμό πορτραίτου.

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
```

Η LoadingActivity είναι η δραστηριότητα εκκίνησης της εφαρμογής .

```
<activity
    android:name="com.example.dietdroid.HelpActivity"
    android:noHistory="true"
    android:screenOrientation="portrait"
    android:stateNotNeeded="true"
    android:launchMode="singleInstance">
</activity>
```

Δεν κρατάμε ιστορικά ίχνη της εφαρμογής , η HelpActivity τελειώνει μόνιμα . Δεν μας ενδιαφέρει να κρατήσουμε την προάγουσα κατάσταση της , σβήνεται από τη στοίβα δραστηριοτήτων. Θέλουμε μόνο

προσανατολισμό πορτραίτου.

```
<activity
    android:name="com.example.dietdroid.SettingsActivity"
    android:screenOrientation="portrait"
    android:noHistory="true"
    android:stateNotNeeded="true"
    android:launchMode="singleInstance" >
</activity>
```

Δεν κρατάμε ιστορικά ίχνη της εφαρμογής , η `SettingsActivity` τελειώνει μόνιμα . Δεν μας ενδιαφέρει να κρατήσουμε την προηγούμενη κατάσταση της , σβήνεται από τη στοίβα δραστηριοτήτων. Θέλουμε μόνο προσανατολισμό πορτραίτου.

```
<activity
    android:name="com.example.dietdroid.FindDietActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Light.NoTitleBar" >
</activity>
```

Χρησιμοποιούμε ένα ξεχωριστό style για αυτήν την activity . Μας ενδιαφέρει να υπάρχει στην στοίβα δραστηριοτήτων , μας ενδιαφέρει και η κατάσταση της (δηλαδή τι είναι γραμμένο σε αυτήν) και να έχει προσανατολισμό πορτραίτου.

```
<activity
    android:name="com.example.dietdroid.ResultsActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Light.NoTitleBar" >
</activity>
```

Χρησιμοποιούμε ένα ξεχωριστό style για αυτήν την activity . Μας ενδιαφέρει να υπάρχει στην στοίβα δραστηριοτήτων , μας ενδιαφέρει και η κατάσταση της (δηλαδή τι είναι γραμμένο σε αυτήν) και να έχει προσανατολισμό πορτραίτου.

```
<activity
    android:name="com.example.dietdroid.MenuActivity"
    android:screenOrientation="portrait"
    android:launchMode="singleInstance"
    android:stateNotNeeded="true" >
</activity>
```

Μας ενδιαφέρει ο προσανατολισμός πορτραίτου σε αυτήν την activity . Η κατάσταση της δεν μας ενδιαφέρει , αλλά θέλουμε να είναι στη στοίβα δραστηριοτήτων για να μπορούμε να επιστρέψουμε σε αυτήν . Επειδή μπορούμε να επιστρέψουμε σε αυτήν από πολλές οθόνες , επιθυμούμε να έχουμε μόνο μια state της activity αυτής .

```
</application>
</manifest>
```

5.2.2 loadinglayout.xml

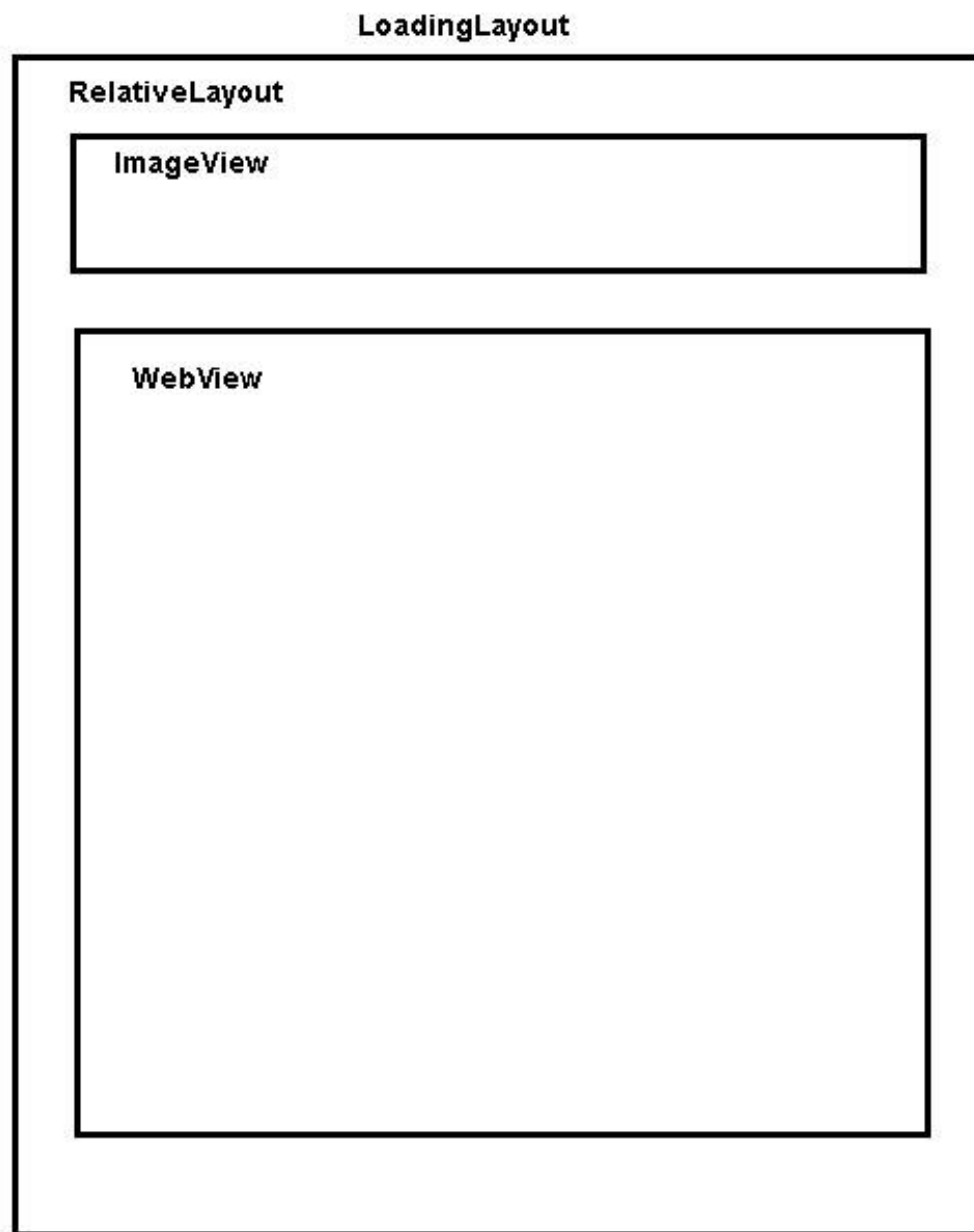
Η κλάση *LoadingActivity.java* αντιπροσωπεύει μια δραστηριότητα της εφαρμογής μας. Ουσιαστικά είναι η πρώτη δραστηριότητα και η πρώτη οθόνη που βλέπουμε. Η κλάση αυτή χρησιμοποιεί το αρχείο *loadinglayout.xml* για να σχηματιστεί η οθόνη της δραστηριότητας. Ο κώδικας του αρχείου αυτού και η απεικόνιση του ακολουθούνε.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/imgV1"
        android:src="@drawable/loading_title"
        android:contentDescription="@string/loading"
        android:layout_alignParentTop="true"/>

    <WebView
        android:id="@+id/wV1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/imgV1" />

</RelativeLayout>
```



Εικόνα 11: loadinglayout.xml

Ακολουθεί η ανάλυση του αρχείου `LoadingActivity.java` που είναι η κύρια κλάση που σχετίζεται με αυτή την `Activity` (δραστηριότητα).

5.2.3 LoadingActivity.java

```
package com.example.dietdroid;
```



```
import java.io.IOException;
import java.io.InputStream;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.webkit.WebView;
```

```
@SuppressWarnings("SetJavaScriptEnabled")
public class LoadingActivity extends Activity {
```

Κατά την εκκίνηση της εφαρμογής (κλικ στο εικονίδιο της εφαρμογής), η πρώτη δραστηριότητα που ενεργοποιείται είναι η *LoadingActivity*. Βρίσκεται μέσα στο πακέτο *com.example.dietdroid*. Σκοπός της δραστηριότητας αυτής είναι να παρουσίαση 2 γραφικά *gif* στο χρήστη.

```
WebView webView_1;
```

Επειδή το android υποστηρίζει τα αρχεία αυτά μέσα από *WebViews* δημιουργούμε μια.

```
@SuppressWarnings("NewApi")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    new FullScreen(this);
    setContentView(R.layout.loadinglayout);
```

Έπειτα δημιουργούμε ένα αντικείμενο της κλάσης *FullScreen* για να καλύψει όλη η δραστηριότητα την οθόνη και ορίζουμε το *layout* της εφαρμογής.

```
webView_1 = (WebView) findViewById(R.id.wV1);
webView_1.getSettings().setJavaScriptEnabled(true);
InputStream insertFile_1 = getResources().openRawResource(
    R.raw.format_loadinglayout);
```

Ενεργοποιούμε την λειτουργίες της Javascript στη συγκεκριμένη WebView και ορίζουμε σε ένα αντικείμενο InputStream τον ακατέργαστο πόρο κειμένου format_loadinglayout.xml .

```
String fileString = null;

try {

    fileString = (new
InputStreamToString(insertFile_1)).sBuffer

        .toString();

} catch (IOException e) {

    e.printStackTrace();

}
```

Με ένα αντικείμενο της βοηθητικής κλάσης InputStreamToString μετατρέπουμε το InputStream αντικείμενο σε ένα αντικείμενο String .

```
webView_1.getSettings().setLoadWithOverviewMode(true);
```

Χρησιμοποιούμε στην WebView τη μέθοδο setLoadWithOverviewMode για να κάνει αυτόματα zoom out και να ταιριάζει το layout ακριβώς στη οθόνη του κινητού.

```
webView_1.getSettings().setUseWideViewPort(true);
```

Χρησιμοποιούμε επίσης και τη μέθοδο setUseWideViewPort με όρισμα true χωρίς όμως να έχουμε ένα tag (width) στα meta data που είναι γραμμένα στο αρχείο format_loadinglayout.xml , οπότε το κινητό χρησιμοποιεί αυτόματα wide viewport για να παρουσίαση το layout.

```
webView_1.loadDataWithBaseURL(null, fileString, "text/html",
"utf-8",

    null);

new JumpToActivity(this, MenuActivity.class, 13000);
```

Τέλος φορτώνουμε το αντικείμενο String στη WebView , με κωδικοποίηση utf-8 , σαν αρχείο text – html , χωρίς κάποιο url (null) ή history entry (null) , και ταυτόχρονα δημιουργούμε αντικείμενο της κλάσης JumpToActivity ώστε να δημιουργήσουν αρχικά μια καθυστέρηση 13 δευτερολέπτων για να εμφανιστούν τα γραφικά και να μεταβαίνουμε στην επόμενη κατά σειρά δραστηριότητα που είναι αυτή του μενού και ταυτόχρονα τερματίζεται η LoadingActivity.

```
new JumpToActivity(this, MenuActivity.class, 2000);

}

public boolean onKeyDown(int keyCode, KeyEvent event) {

if (keyCode == KeyEvent.KEYCODE_BACK) {
```

```
        new ExitDroid>LoadingActivity.this);  
    }  
    return false;  
}  
}
```

Αξίζει να σημειωθεί ότι έχει προστεθεί και η δυνατότητα εξόδου από την εφαρμογή άμα το απόφαση ο χρήστης , καθώς με τη συνάρτηση `onKeyDown` , πατώντας ο χρήστης το `back Button` ενεργοποιεί ένα αντικείμενο `ExitDroid` που δημιουργεί ένα παράθυρο διαλόγου και ρωτά το χρήστη για πιθανή έξοδο.

Πανεπιστήμιο Πειραιώς

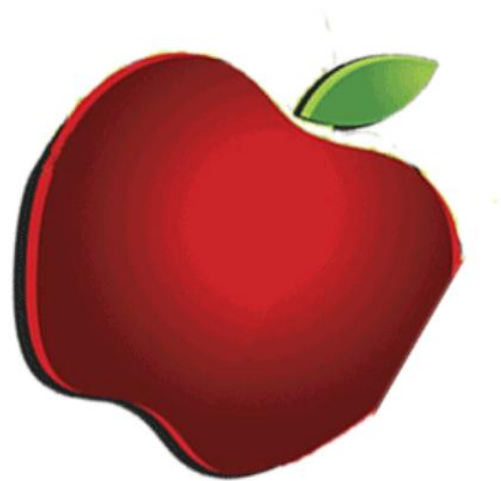
Loading screen



Loadi

Εικόνα 12: Loading Screen (1)

Loading screen



Loading...

Εικόνα 13: Loading Screen (2)

Loading screen



Εικόνα 14: Loading Screen (3)

Loading screen



Loading...

Εικόνα 15: Loading Screen (4)

5.2.4 menulayout.xml

Η δραστηριότητα αυτή είναι ουσιαστικά αυτή του μενού που βλέπει ο χρήστης . Η εφαρμογή μας μεταβαίνει εκεί αναγκαστικά μετά την οθόνη φόρτωσης . Όπως δείχνει και το διάγραμμα δραστηριοτήτων - οθονών της εφαρμογής μπορούμε να μεταβούμε στην οθόνη αυτή μόνο από την οθόνη φόρτωσης ή με πίσω κίνηση (*back button*) από τις οθόνες που μπορούμε να μεταβούμε από το μενού. Το *interface* της οθόνης χρησιμοποιεί το πόρο *layout* , *menulayout.xml* και επιπλέον η *ListView* που περιέχει έχει δικό της πόρο *layout* , *menulayoutitem.xml* , για το κάθε *item* της λίστας . Τέλος , ένα δεύτερο υπομενού με τις ίδιες επιλογές με το αρχικό εμφανίζεται όταν ο χρήστης πατήσει το κουμπί του μενού της συσκευής του . Σε εκείνη την επιλογή χρησιμοποιείται ο *layout* πόρος , *menu_mainmenu.xml* . Το βασικό μας μενού ονομάζεται μενού περιβάλλοντος και το δεύτερο (υπομενού) μενού επιλογών.

Μενού περιβάλλοντος είναι το μενού που εμφανίζεται , όταν ο χρήστης κάνει ένα μακρόσυρτο κλικ σε ένα αντικείμενο *View* . Αυτός ο τύπος μενού χρησιμοποιείται συχνά σε συνδυασμό με μηχανισμούς ελέγχου *ListView* που είναι συμπληρωμένοι με παρόμοια στοιχεία.

Μενού επιλογών είναι το μενού που εμφανίζεται όταν ένας χρήστης κάνει κλικ στο κουμπί μενού της συσκευής του . Αυτός ο τύπος μενού χρησιμοποιείται συχνά για να βοηθήσει τον χρήστη να χειριστεί ρυθμίσεις εφαρμογής και παρόμοια πράγματα.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/imgV2"
        android:src="@drawable/menu_title"
        android:contentDescription="@string/menu"
        android:layout_alignParentTop="true"/>

    <ListView
        android:background="@drawable/background"
        android:layout_centerVertical="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerHorizontal="true"
        android:id="@+id/ListV1"
        android:layout_below="@+id/imgV2"
        android:divider="@drawable/menu_divider"
        android:listSelector="@drawable/menu_selector">
    </ListView>
</RelativeLayout>
```


5.2.5 menulayoutitem.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:textColor="@color/menulayoutitem_textcolor"
    android:textSize="@dimen/menulayoutitem_textsize"
    android:shadowColor="@color/menulayoutitem_shadowcolor"
    android:shadowDx="3"
    android:shadowDy="3"
    android:shadowRadius="5"
    android:gravity="center"
    android:layout_gravity="center_horizontal|center_vertical"
    android:background="@color/menulayoutitem_background"
    android:id="@+id/textV1">

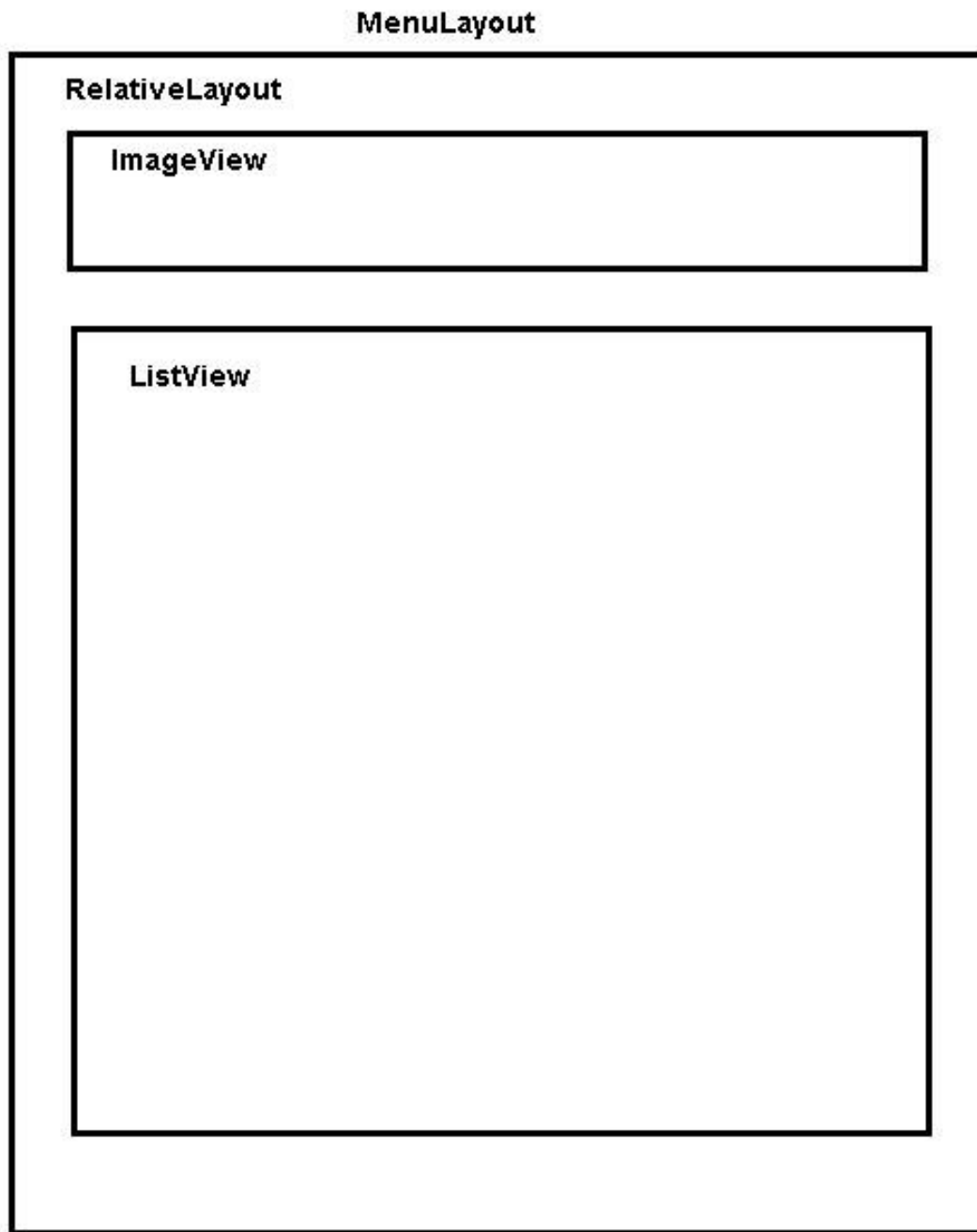
</TextView>
```

Ο παραπάνω κώδικας χρησιμοποιείται για την μορφοποίηση του κάθε ξεχωριστού item του menu στην MenuScreen .

5.2.6 menu_mainmenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/find_diet_menu_item"
        android:title="@string/find_diet_item"
        android:icon="@drawable/finddiet_icon"></item>
    <item
        android:id="@+id/settings_menu_item"
        android:title="@string/settings_item"
        android:icon="@drawable/settings_menu_icon"></item>
    <item
        android:id="@+id/help_menu_item"
        android:title="@string/help_item"
        android:icon="@drawable/help_menu_icon"></item>
    <item
        android:id="@+id/exit_menu_item"
        android:title="@string/exit_item"
        android:icon="@drawable/exit_menu_icon"></item>
</menu>
```

Ο παραπάνω κώδικας χρησιμοποιείται για την μορφοποίηση του κάθε ξεχωριστού item του menu επιλογών της συσκευής στην MenuScreen .



Εικόνα 16: menulayout.xml

Ακολουθεί η ανάλυση της κλάσης `MenuActivity.java` που σχετίζεται με την οθόνη – δραστηριότητα του μενού της εφαρμογής.

5.2.7 `MenuActivity.java`

Η κλάση αυτή είναι πολύ σημαντική για την εφαρμογή μας . Πέρα από την υλοποίηση του μενού περιβάλλοντος και του μενού επιλογών , που χρησιμοποιούνται για να μεταβούμε σε άλλες Ανάπτυξη Android Εφαρμογής «Δίαιτα (DietDroid)»

δραστηριότητες – οθόνες , η κλάση αυτή έχει και άλλη χρήσιμη λειτουργία. Όταν εγκαταστήσουμε για πρώτη φορά την εφαρμογή μας και την τρέξουμε , όταν αυτή φτάσει στο μενού για πρώτη φορά δημιουργείται η βάση μας . Όπως αναφέρθηκε και παραπάνω , η βάση μας είναι στατική , δηλαδή δημιουργείται μια φορά , μια φορά γίνονται εγγραφές και δεν γίνονται *updates* . Μόνο αναζητήσεις μπορεί να κάνει κανείς στη βάση μας .

```
package com.example.dietdroid;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class MenuActivity extends Activity {

    SharedPreferences sharedPreferences;
    DatabaseHandler database;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        new FullScreen(this);
        setContentView(R.layout.menulayout);
    }
}
```

Η κλάση αυτή ανήκει με τη σειρά της στο ίδιο πακέτο και με τις άλλες , *com.example.android* και όταν καλείται δημιουργεί ένα αντικείμενο *FullScreen* , για να δείχνει το *layout* που χρησιμοποιεί (*menulayout*) σε όλη την οθόνη .

```
        sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this);

        if (sharedPreferences.getString("set_pref",
"default").toString()
                .equals("default")) {
```

```

        setFirstTimePreferences();
    }

```

Εδώ εξετάζεται αν οι προτιμήσεις του χρήστη είναι ίδιες με τις default , πράγμα που σημαίνει ότι ο πρέπει να καλέσουμε την setFirstTimePreferences() και στη συνέχεια να δημιουργηθεί για πρώτη φορά η βάση μας.

```

        database = new DatabaseHandler(this);
        if (sharedPreferences.getString("create_base",
"created").toString()
            .equals("not_created")) {
            insert_data(database);

            Editor editor = sharedPreferences.edit();
            editor.putString("create_base", "created");
            editor.commit();
        }

```

Με τον παραπάνω κώδικα εξετάζεται αν έχει δημιουργηθεί ξανά η βάση μας . Αν όχι καλούμε την insert_data() , την δημιουργούμε και τις προσθέτουμε τιμές και έπειτα αναγκαστικά για να μην ξαναγίνει η ίδια διαδικασία στο μέλλον , αλλάζουμε με την βοήθεια ενός Editor τη flag τιμή που έχουμε βάλει στις προτιμήσεις για την δημιουργία της βάσης.

```

        ListView listView_1 = (ListView) findViewById(R.id.listView1);
        String[] listView_1_items = {
            getResources().getString(R.string.find_diet_item),
            getResources().getString(R.string.settings_item),
            getResources().getString(R.string.help_item),
            getResources().getString(R.string.exit_item) };

```

Ανακτούμε ένα μηχανισμό ελέγχου *ListView* με το μοναδικό αναγνωριστικό του (*id*) και στη συνέχεια ορίζουμε τιμές *String* που θα χρησιμοποιηθούν για να συμπληρώσουμε τα στοιχεία *TextView* μέσα στο μηχανισμό *ListView*.

```

        ArrayAdapter<String> adapter_1 = new ArrayAdapter<String>(
            MenuActivity.this, R.layout.menulayoutitem,
            listView_1_items);
        listView_1.setAdapter(adapter_1);

```

Έχουμε ανακτήσει τον μηχανισμό *ListView* , έχουμε και τα δεδομένα για να τον συμπληρώσουμε και χρησιμοποιούμε ένα προσαρμογέα δεδομένων , ώστε να αντιστοιχίσουμε τα δεδομένα με το πρότυπο διάταξης (*menulayoutitem.xml*) . Η επιλογή προσαρμογέα εξαρτάται από τον τύπο των δεδομένων που χρησιμοποιούνται . Στην περίπτωση μας *ArrayAdapter*.

```

        listView_1
            .setOnClickListener(new
AdapterView.OnItemClickListener() {

                @Override
                public void onItemClick(AdapterView<?>
parent,
                                View itemClicked, int position,
long id) {

                                TextView textView_1 = (TextView)
itemClicked;
                                String strText_1 =
textView_1.getText().toString();

                                if
(strText_1.equalsIgnoreCase(getResources()
                                .getString(R.string.find_diet_item))) {
                                // Launch the Find Diet
Activity
                                startActivity(new
Intent(MenuActivity.this,
                                FindDietActivity.class));
                                } else if
(strText_1.equalsIgnoreCase(getResources()
                                .getString(R.string.help_item))) {
                                // Launch the Help Activity
startActivity(new
Intent(MenuActivity.this,
                                HelpActivity.class));
                                } else if
(strText_1.equalsIgnoreCase(getResources()
                                .getString(R.string.settings_item))) {
                                // Launch the Settings Activity
startActivity(new
Intent(MenuActivity.this,
                                SettingsActivity.class));
                                } else if
(strText_1.equalsIgnoreCase(getResources()

```

```

        .getString(R.string.exit_item))) {
                                // Exit Droid
                                new
ExitDroid(MenuActivity.this);
                                }
        }
    });
}

```

Στον παραπάνω κώδικα ουσιαστικά υλοποιείται ένας *listener* πάνω στο μηχανισμό *ListView*. Η μέθοδος *onItemClick()* μεταβιβάζει όλες τις πληροφορίες που χρειάζονται για να καθορίσουν σε ποιο στοιχείο έγινε το κλικ. Για να δούμε σε ποια δραστηριότητα (ή έξοδο) θα εκκινήσει ελέγχουμε αν το κείμενο του μηχανισμού *View* που έγινε το κλικ, είναι ίδιο με το «*Find Diet*», «*Help*», «*Settings*» και «*Exit*» και μεταβαίνουμε στην αντίστοιχη οθόνη – δραστηριότητα.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.menu_mainmenu, menu);
    menu.findItem(R.id.find_diet_menu_item).setIntent(
        new Intent(MenuActivity.this,
FindDietActivity.class));
    menu.findItem(R.id.settings_menu_item).setIntent(
        new Intent(MenuActivity.this,
SettingsActivity.class));
    menu.findItem(R.id.help_menu_item).setIntent(
        new Intent(MenuActivity.this, HelpActivity.class));
    // sto exit itam tou menu den bazoume gia na mporoume na
bgainoume kanw
    // prwta to if...
    return true;
}

```

Με την μέθοδο *onCreateOptionsMenu(Menu menu)* υλοποιούμε το μενού επιλογών της εφαρμογής μας. Στο μενού επιλογών φορτώνουμε ουσιαστικά ένα πόρο διάταξης μενού (*menu_mainmenu.xml*), και θέτουμε τις κατάλληλες πληροφορίες *Intent* για το κάθε στοιχείο του μενού.

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);
    String strMenuItem_1 = item.getTitle().toString();
    if (strMenuItem_1.equalsIgnoreCase(getResources().getString(
        R.string.exit_item))) {

```

```

        new ExitDroid(MenuActivity.this);
    } else {
        startActivity(item.getIntent());
    }
    return true;
}

```

Για να κάνουμε ακρόαση όταν ο χρήστης εκκινεί το μενού επιλογών και επιλέγει μια επιλογή του μενού , υλοποιούμε τη μέθοδο *onOptionsItemSelected()* . Έτσι μπορούμε να αρχίσουμε την κατάλληλη δραστηριότητα – οθόνη εξάγοντας την πρόθεση από το στοιχείο του μενού που επέλεξε ο χρήστης.

```

public void setFirstTimePreferences() {
    Editor editor = sharedPreferences.edit();
    editor.putString("set_pref", "not_default");
    editor.putString("max_weight", "200");
    editor.putString("max_height", "200");
    editor.putString("max_age", "100");
    editor.putString("create_base", "not_created");
    editor.commit();
}

```

Με τη μέθοδο *setFirstTimePreferences()* ορίζουμε για πρώτη φορά τις προτιμήσεις του χρήστη στη εφαρμογή μας . Είναι ουσιαστικά οι *default* προτιμήσεις της εφαρμογής . Μέσα στις προτιμήσεις έχει δημιουργηθεί και ένα *flag* – προτίμηση , που μας ενημερώνει αν δημιουργήθηκε παλαιότερα η βάση μας . Οι τιμές των προτιμήσεων αλλάζουν από τα *settings* ενώ η τιμή της *flag* – προτιμήσεις (*create_base*) αλλάζει όταν δημιουργείται η βάση και ποτέ ξανά . Άρα τη δεύτερη φορά που μπαίνει ο χρήστης στη δραστηριότητα – οθόνη μενού , η βάση δεν δημιουργείται ξανά , και σαν τιμές των *settings* είναι αυτές που έβαλε τελευταία φορά.

```

public void insert_data(DatabaseHandler db) {
    // Inserting
    Log.d("Insert: ", "Inserting ..");
}

```

Υλοποιείται η μέθοδος *insert_data(DatabaseHandler db)* για να προσθέσουμε στοιχεία στη βάση μας.

```

db.addPrwina(new Prwina("Τοστ με αβγό", 255));
. . . . .

```

Προσθέτουμε στη βάση μας , στον πίνακα με τα πρωινά , 50 εγγραφές με τις ονομασίες πρωινών και τις θερμίδες που έχουν. Ο κώδικας εισαγωγής στοιχείων στη βάση μας βρίσκεται στο παράρτημα . Τα στοιχεία είναι αληθινά και προέρχονται από το *site* :

<http://www.atcare.gr> .

```

db.addMesimeriana(new Mesimeriana("Αμπελοφάσουλα σαλάτα", 340));
. . . . .

```

Προσθέτουμε στη βάση μας, στον πίνακα με τα μεσημεριανά, 50 εγγραφές με τις ονομασίες μεσημεριανών και τις θερμίδες που έχουν. Ο κώδικας εισαγωγής στοιχείων στη βάση μας βρίσκεται στο παράρτημα. Τα στοιχεία είναι αληθινά και προέρχονται από το *site* : <http://www.atcare.gr>.

```
db.addBradyna(new Bradyna("Αχλάδι κομπόστα (1 μέτριο)", 165));
. . . . .
```

Προσθέτουμε στη βάση μας, στον πίνακα με τα βραδινά, 50 εγγραφές με τις ονομασίες βραδινών και τις θερμίδες που έχουν. Ο κώδικας εισαγωγής στοιχείων στη βάση μας βρίσκεται στο παράρτημα. Τα στοιχεία είναι αληθινά και προέρχονται από το *site* :

<http://www.atcare.gr>.

```
public boolean onKeyDown(int keyCode, KeyEvent event) {

    if (keyCode == KeyEvent.KEYCODE_BACK) {
        new ExitDroid(MenuActivity.this);
    }
    return false;
}
}
```

Με την παραπάνω μέθοδο εξασφαλίζουμε ότι αν ο χρήστης πατήσει το *Back Button* στο μενού θα του εμφανιστεί η επιλογή της εξόδου από την εφαρμογή.

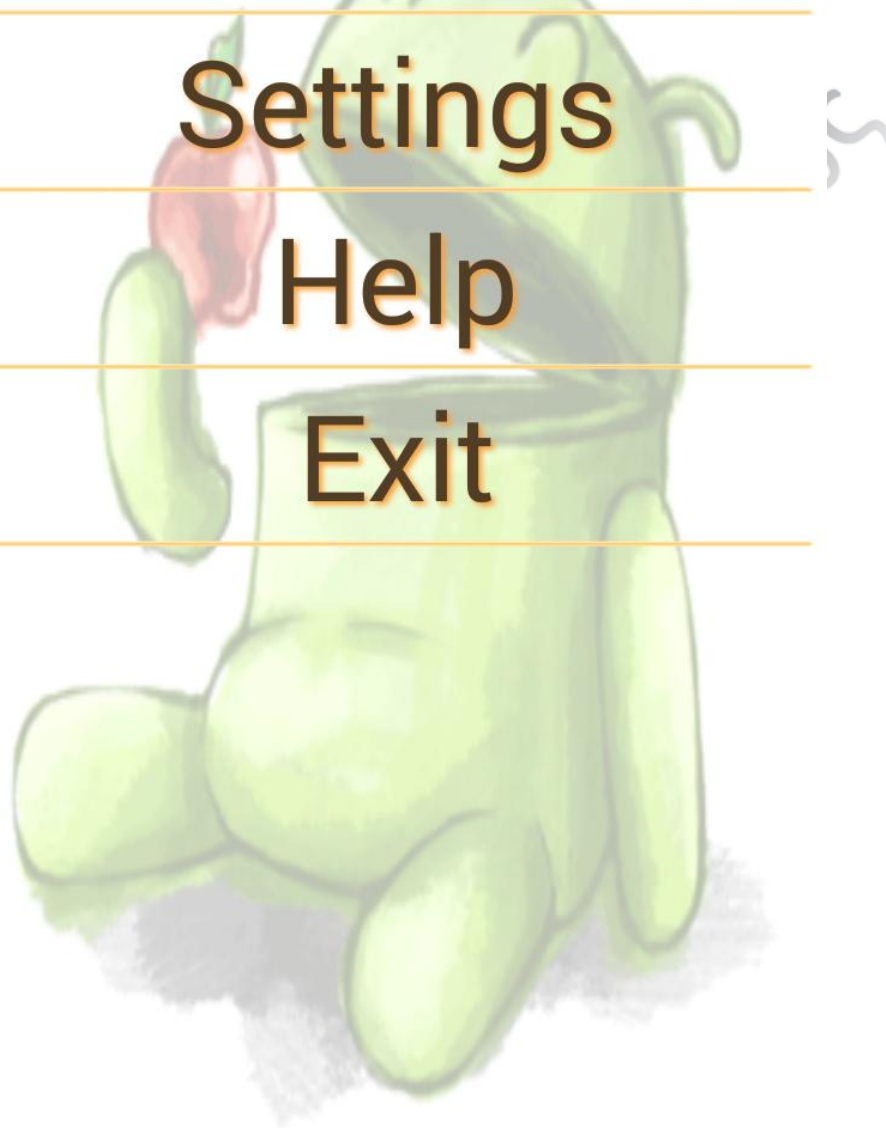
MENU SCREEN

Find Diet

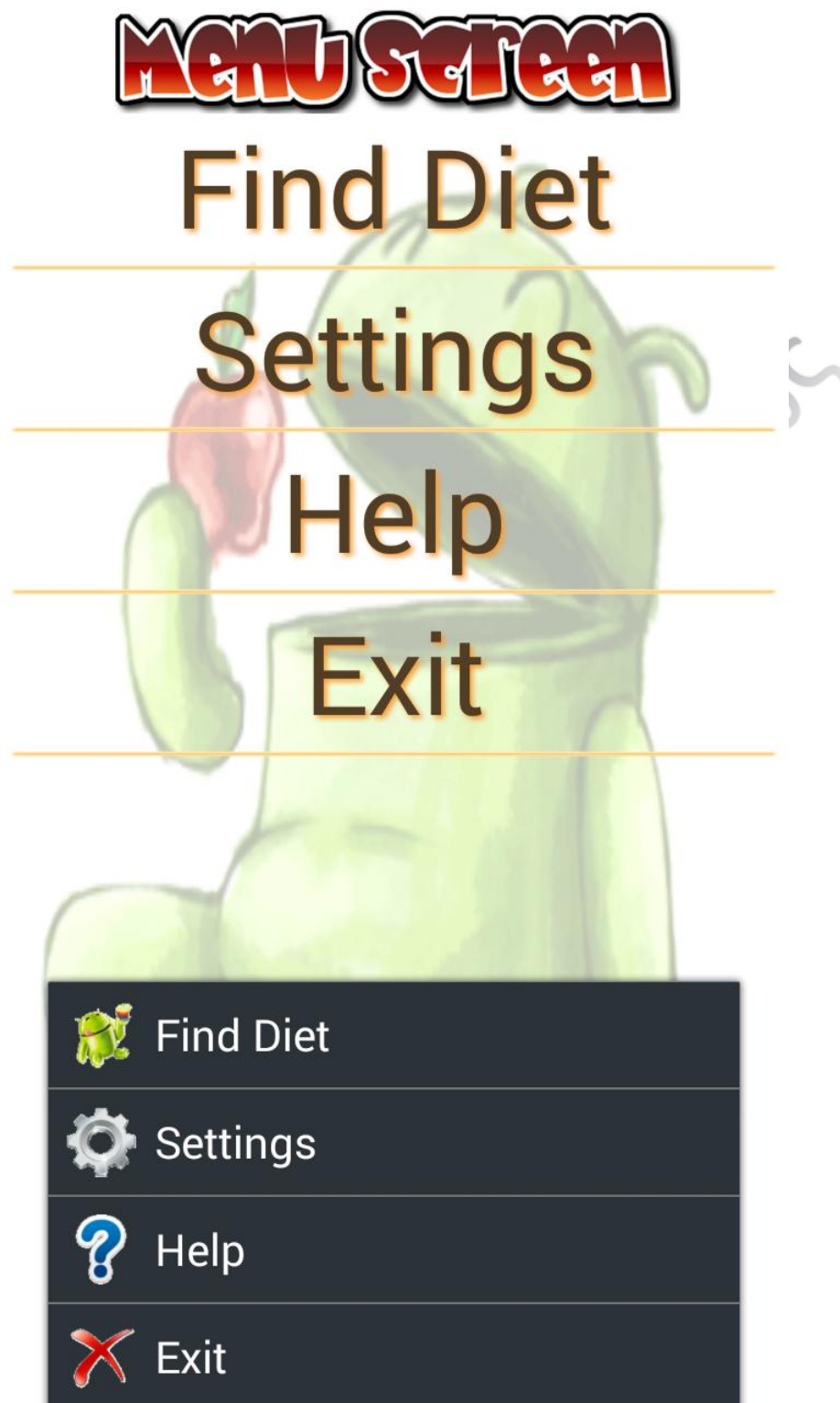
Settings

Help

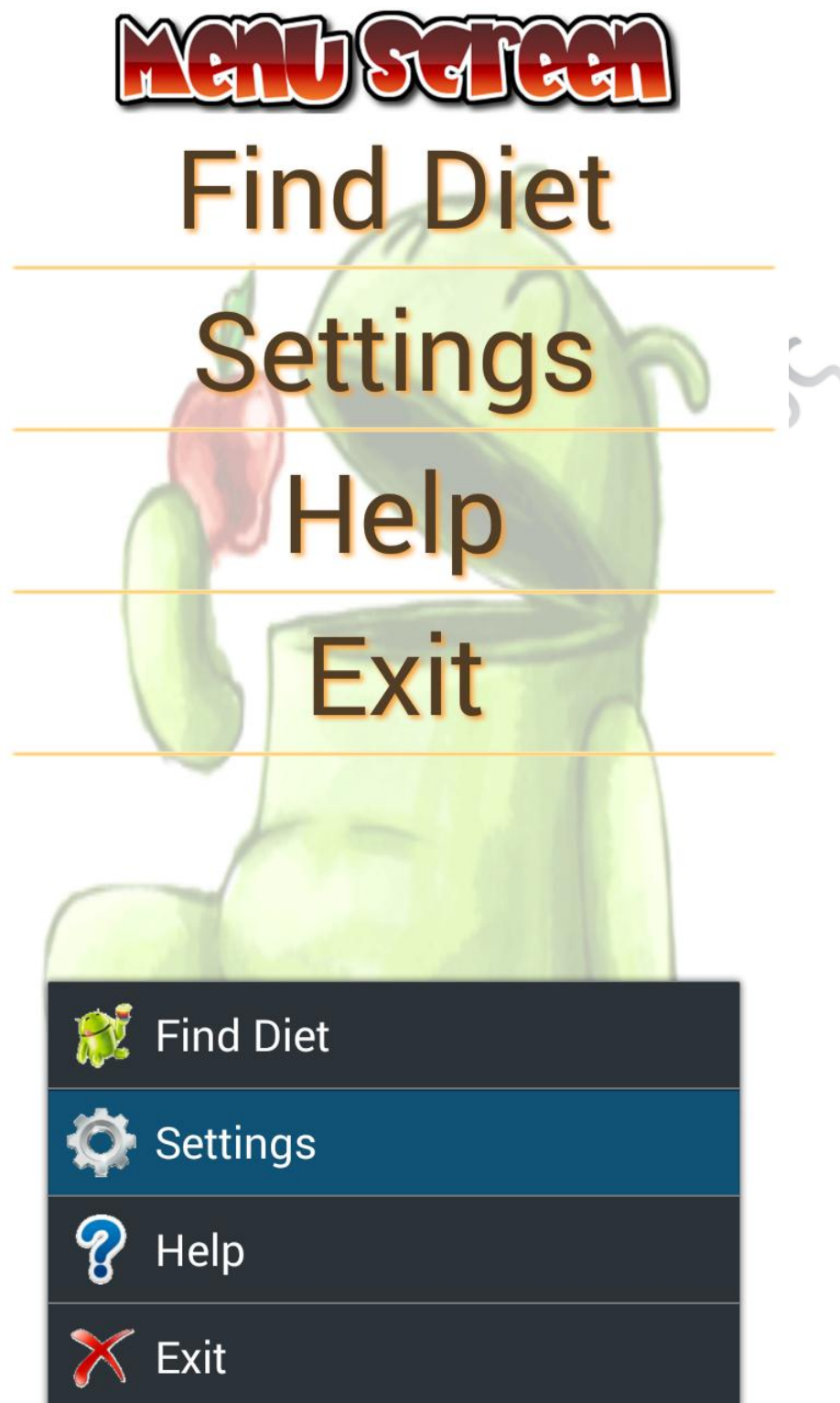
Exit



Εικόνα 17: Menu Screen (1)



Εικόνα 18: Mobile Menu (1)



Εικόνα 19: Mobile Menu(2)

5.2.8 helplayout.xml

Η δραστηριότητα αυτή αφορά την οθόνη της βοήθειας της εφαρμογής . Μπορείς να έχεις πρόσβαση σε αυτή με 2 τρόπους και οι 2 από το μενού . Ο ένας τρόπος με απευθείας επιλογή από το μενού και ο άλλος από το αναδυόμενο μενού , αφού πρώτα ο χρήστης πατήσει το κουμπί *menu* του κινητού . Η δραστηριότητα της βοήθειας χρησιμοποιεί το πόρο *helplayout.xml* με το παρακάτω κώδικα και απεικόνιση.

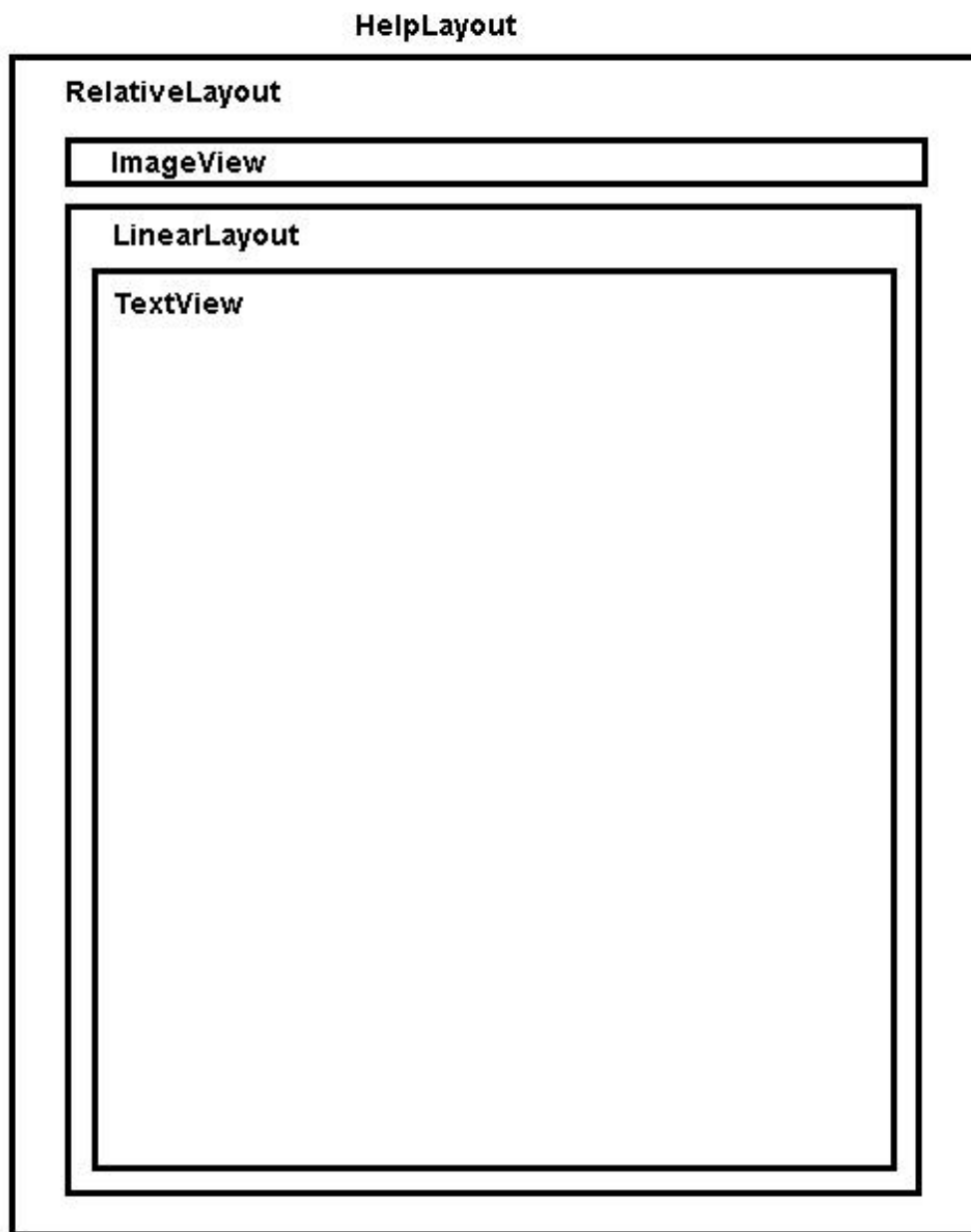
```
<?xml version="1.0" encoding="UTF-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_gravity="center_vertical">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/imgV3"
        android:src="@drawable/help_title"
        android:contentDescription="@string/help"
        android:layout_alignParentTop="true"/>

    <LinearLayout
        android:layout_below="@+id/imgV3"
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/helplayout_background">

        <TextView
            android:id="@+id/textV2"
            android:background="@color/helplayout_textview_background"
            android:textColor="@color/helplayout_textcolor"
            android:textStyle="italic"
            android:textSize="@dimen/helplayout_textsize"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:paddingRight="@dimen/helplayout_textpadding_r"
            android:paddingLeft="@dimen/helplayout_textpadding_l"
            android:paddingBottom="@dimen/helplayout_textpadding_t_b"
            android:paddingTop="@dimen/helplayout_textpadding_t_b"
            android:isScrollContainer="true"
            android:scrollbarStyle="outsideOverLay"
            android:scrollbars="vertical"/>

    </LinearLayout>
</RelativeLayout>
```



Εικόνα 20: helplayout.xml

Αξίζει να σημειωθεί ότι μέσα στο παραπάνω *TextView* έχει ενσωματωθεί *scrollbars* , πράγμα που μπορούσε να γίνει τοποθετώντας μια *ScrollView* μέσα σε *RelativeLayout* και αυτό μέσα στο *TextView* . Ακολουθεί η ανάλυση του αρχείου *HelpActivity.java* που είναι η κύρια κλάση που σχετίζεται με αυτή την *Activity* (δραστηριότητα).

5.2.9 HelpActivity.java

```
package com.example.dietdroid;
```

```
import java.io.IOException;
import java.io.InputStream;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;

public class HelpActivity extends Activity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        new FullScreen(this);
        setContentView(R.layout.helplayout);
    }
}
```

Η κλάση μας ανήκει και αυτή στο πακέτο *com.example.dietdroid* και έχει χρησιμοποιηθεί η κλάση *FullScreen* για να καλύπτει το χρησιμοποιούμενο *layout (helplayout)* ολόκληρη την οθόνη.

```
InputStream insertFile_2 =
getResources().openRawResource(R.raw.help_text);

String fileString = null;

try {

    fileString = (new
InputStreamToString(insertFile_2)).sBuffer.toString();

} catch (IOException e) {

    e.printStackTrace();

}
```

Χρησιμοποιούμε να αντικείμενο *InputStream* και του αναθέτουμε το αρχείο ακατέργαστου πόρου *help_text.txt* και ταυτόχρονα χρησιμοποιούμε ένα αντικείμενο από την βοηθητική μας κλάση *InputStreamToString* για να ανοίξουμε και να διαβάσουμε το αρχείο αυτό και να αναθέσουμε το περιεχόμενο του σε ένα αντικείμενο *String*.

```
TextView textView_1 = (TextView) findViewById(R.id.textV2);  
textView_1.setText(fileString);  
textView_1.setMovementMethod(new ScrollingMovementMethod());  
}  
}
```

Το αντικείμενο *String* ορίζεται ως κείμενο στο *TextView* μας και προστίθενται και στο *TextView* οι *ScrollBars* για κύλιση του μενού.

MENU SCREEN

Find Diet

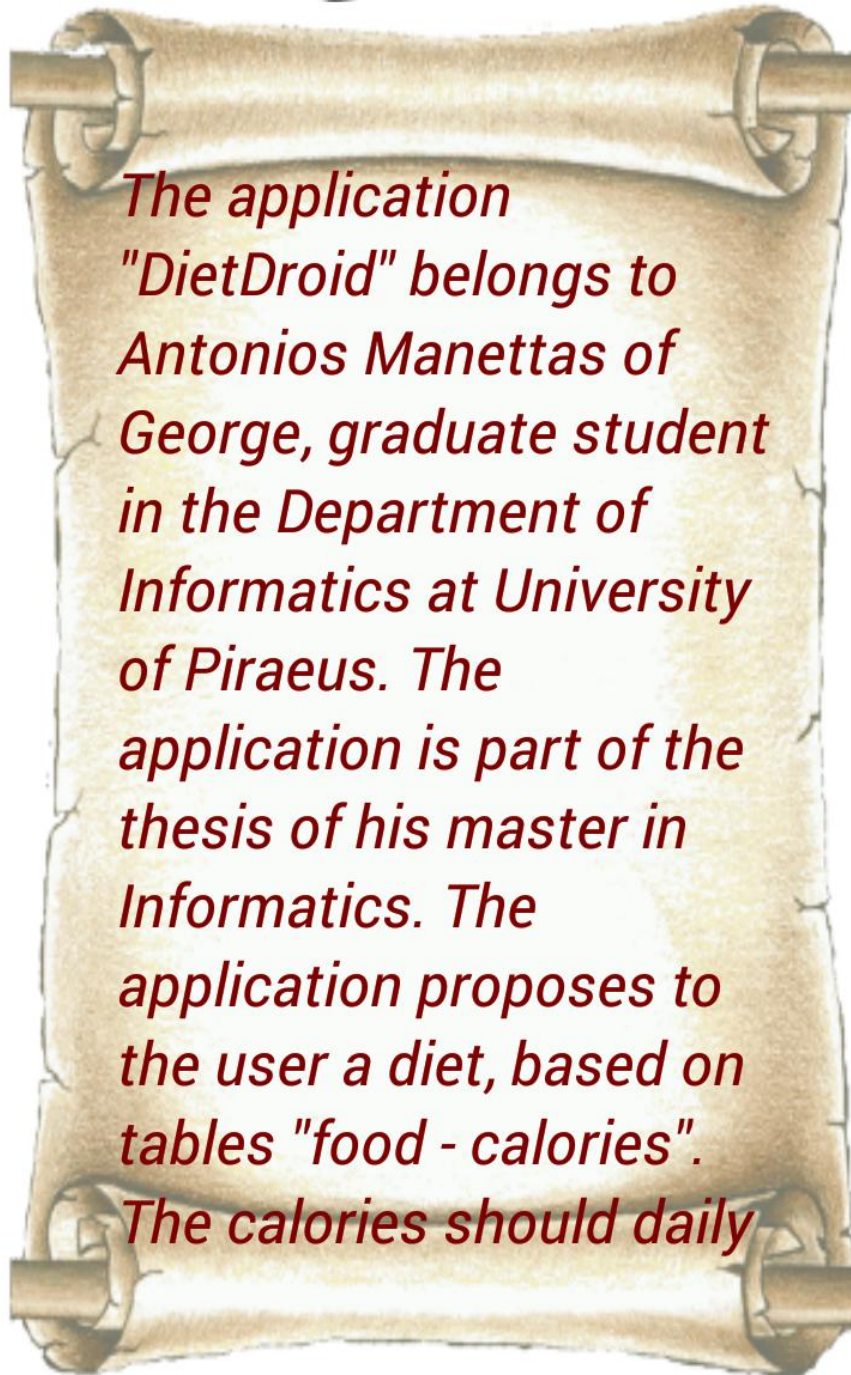
Settings

Help

Exit

Εικόνα 21: MenuScreen (2)

HELP SCREEN



Εικόνα 22: Help Screen (1)

HELP SCREEN

application proposes to the user a diet, based on tables "food - calories". The calories should daily take the user resulting from his "BMI". The "BMI" has several parameters, such as weight, height, sex, etc. and the calculation is based on truthful data supplied by the user in the application.

Εικόνα 23: Help Screen (2)

5.2.10 settinglayout.xml

Η δραστηριότητα αυτή αφορά την οθόνη των ρυθμίσεων της εφαρμογής . Μπορείς να έχεις πρόσβαση σε αυτή με 2 τρόπους και οι 2 από το μενού . Ο ένας τρόπος με απευθείας επιλογή από το μενού και ο άλλος από το αναδυόμενο μενού , αφού πρώτα ο χρήστης πατήσει το κουμπί menu του κινητού . Η δραστηριότητα της βοήθειας χρησιμοποιεί το πόρο *settinglayout.xml* με το παρακάτω κώδικα και απεικόνιση.

```
<?xml version="1.0" encoding="UTF-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imgV4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:contentDescription="@string/settings"
        android:src="@drawable/settings_title" />

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@+id/imgV4"
        android:background="@drawable/background" >

        <ScrollView
            android:id="@+id/scroll3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:background="@android:color/transparent" >

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@android:color/transparent" >

                <TextView
                    android:id="@+id/textV18"
                    style="@style/textviews_settings"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@string/megisto_baros" />

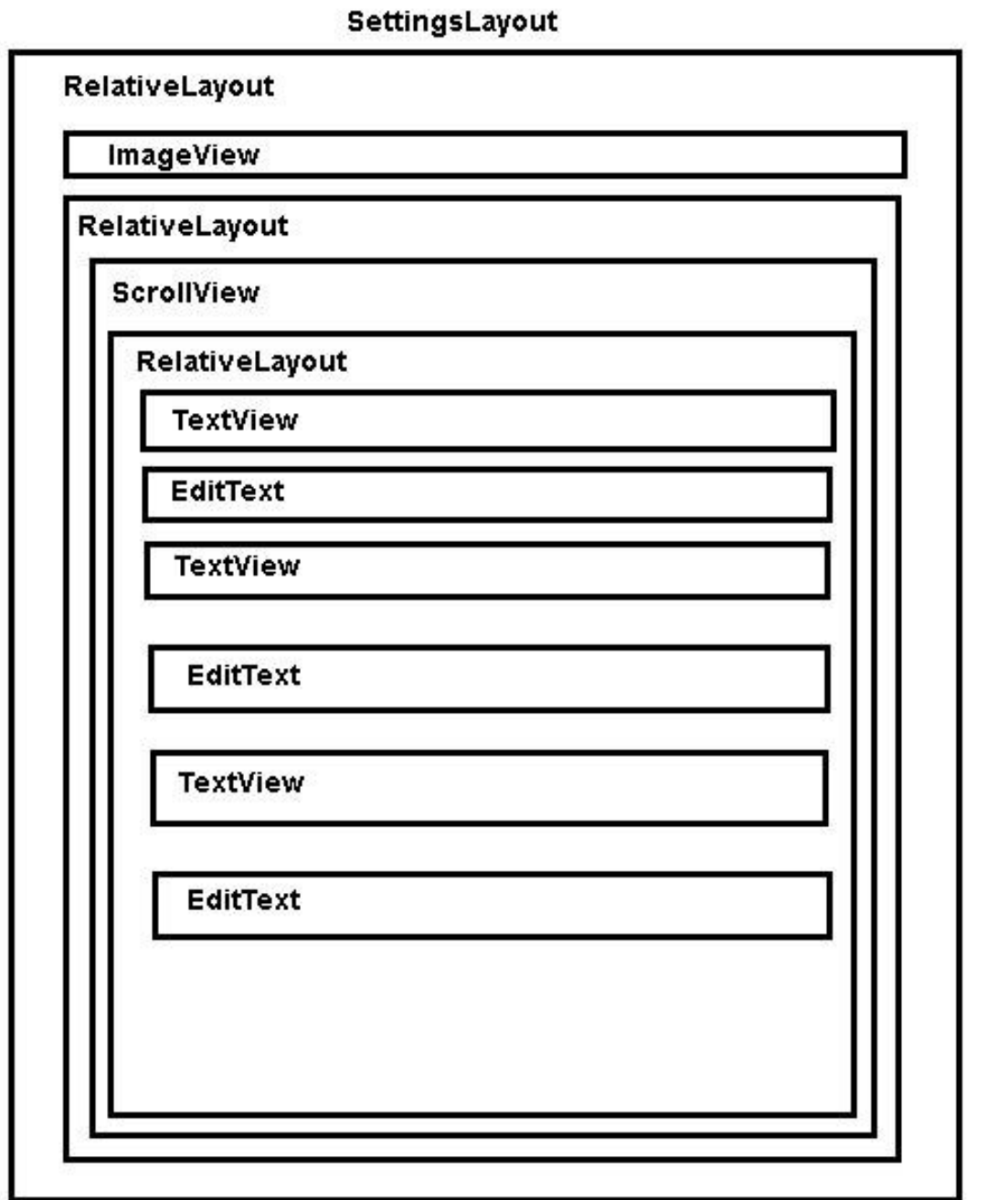
                <EditText
                    android:id="@+id/editT9"
                    style="@style/edittexts_settings"
                    android:layout_below="@+id/textV18" >
            </EditText>
        </ScrollView>
    </RelativeLayout>
</RelativeLayout>
```

```
<TextView
    android:id="@+id/textV19"
    style="@style/textviews_settings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editT9"
    android:text="@string/megisto_ypsos" />

<EditText
    android:id="@+id/editT10"
    style="@style/edittexts_settings"
    android:layout_below="@+id/textV19" >
</EditText>

<TextView
    android:id="@+id/textV20"
    style="@style/textviews_settings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editT10"
    android:text="@string/megisto_ilikia" />

<EditText
    android:id="@+id/editT11"
    style="@style/edittexts_settings"
    android:layout_below="@+id/textV20" >
</EditText>
</RelativeLayout>
</ScrollView>
</RelativeLayout>
</RelativeLayout>
```



Εικόνα 24: settinglayout.xml

Κάτι σημαντικό που πρέπει να επισημάνουμε είναι ότι στα παραπάνω πεδία εισαγωγής κειμένου (*EditText*) υπάρχει ο περιορισμός μέσα στο layout να εμφανίζονται στο πληκτρολόγιο του κινητού μόνο ψηφία , όχι γράμματα , καθώς και η έκταση εισαγωγής να είναι 3ψηφιο νούμερο (καθώς είναι παράλογο ύψος , βάρος και ηλικία να είναι περισσότερα από 3 ψηφία).

Ακολουθεί η ανάλυση του αρχείου *SettingsActivity.java* που είναι η κύρια κλάση που σχετίζεται με αυτή την Activity (δραστηριότητα). Κύρια λειτουργία της κλάσης αυτής είναι ρύθμιση από τον χρήστη τριών μεταβλητών (μέγιστο βάρος , μέγιστο ύψος , μέγιστη ηλικία) , οι οποίες χρησιμεύουν σαν

όρια στη *FindDietActivity* και οι όποιες αποθηκεύονται και μετά την έξοδο του χρήστη από την εφαρμογή, σαν προτιμήσεις του χρήστη (*SharedPreferences*).

5.2.11 **SettingsActivity.java**

```
package com.example.dietdroid;

import android.app.Activity;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;

public class SettingsActivity extends Activity {

    EditText editText_9, editText_10, editText_11 ;
    SharedPreferences sharedPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        new FullScreen(this);
        setContentView(R.layout.settingslayout);
```

Η κλάση αυτή ανήκει στο πακέτο *com.example.dietdroid*. Καλεί αρχικά τη βοηθητική κλάση *FullScreen* για να καλύπτεται όλη η οθόνη του κινητού από το *layout* που ενσωματώνεται (*settinglayout.xml*).

```
        editText_9 = (EditText) findViewById(R.id.editT9);
        editText_10 = (EditText) findViewById(R.id.editT10);
```

```

        editText_11 = (EditText) findViewById(R.id.editT11);

        sharedPreferences =
PreferenceManager.getDefaultSharedPreferences(this);

        loadSavedPreferences();

```

Ανοίγουμε τον PreferenceManager με τις default προτιμήσεις και μετά (αν και εφόσον υπάρχουν – την πρώτη φορά χρήσης της εφαρμογής οι default θα ήταν αυτές που φορτώνονταν) φορτώνουμε και διαχειριζόμαστε τις προτιμήσεις που είχε ο χρήστης ρυθμίσει σε προηγούμενη χρήση της εφαρμογής.

```

        editText_9.addTextChangedListener(new TextWatcher() {

            @Override

            public void afterTextChanged(Editable arg0) {

```

Βάζουμε στο πεδίο εισαγωγής κειμένου (EditText) ένα listener αλλαγής κειμένου για να ελέγχουμε αν και πως μεταβάλλονται τα στοιχεία του πεδίου από τον χρήστη.

```

                if
(! (editText_9.getText().toString().equals("")) && (arg0.charAt(0) != '0')) {

```

Βάζουμε έλεγχο στο πεδίο για να μην μπορεί ο χρήστης ο να αφήσει κενό το πεδίο αυτό και για να μην μπορεί να βάλει αριθμούς που αρχίζουμε με το ψηφίο μηδέν(0) , όπως είναι το 001 , 011, 000 .

```

                savePreferences("max_weight", editText_9.getText()

                    .toString());

            }

            editText_9.setSelection(editText_9.length());

        }

```

Σώζουμε το πεδίο στις προτιμήσεις του χρήστη και χρησιμοποιούμε τη φράση *setSelection* για να πάει ο *cursor* εισαγωγής στο τελευταίο ψηφίο.

```

            @Override

            public void beforeTextChanged(CharSequence s, int start,

int count,

                int after) {

            }

```

```

        @Override
        public void onTextChanged(CharSequence s, int start, int
before,
                int count) {
        }
    });
    editText_10.addTextChangedListener(new TextWatcher() {

        @Override
        public void afterTextChanged(Editable arg0) {

```

Βάζουμε στο πεδίο εισαγωγής κειμένου (*EditText*) ένα *listener* αλλαγής κειμένου για να ελέγχουμε αν και πως μεταβάλλονται τα στοιχεία του πεδίου από τον χρήστη.

```

        if
(! (editText_10.getText().toString().equals("")) && (arg0.charAt(0) != '0')) {

```

Βάζουμε έλεγχο στο πεδίο για να μην μπορεί ο χρήστης ο να αφήσει κενό το πεδίο αυτό και για να μην μπορεί να βάλει αριθμούς που αρχίζουμε με το ψηφίο μηδέν(0) , όπως είναι το 001 , 011, 000 .

```

        savePreferences("max_height", editText_10.getText()
                .toString());
    }
    editText_10.setSelection(editText_10.length());

```

Σώζουμε το πεδίο στις προτιμήσεις του χρήστη και χρησιμοποιούμε τη φράση *setSelection* για να πάει ο *cursor* εισαγωγής στο τελευταίο ψηφίο.

```

    }

    @Override
    public void beforeTextChanged(CharSequence s, int start,
int count,
                int after) {
    }

```



```

@Override
public void onTextChanged(CharSequence s, int start, int
before,
int count) {
}
});
editText_11.addTextChangedListener(new TextWatcher() {

@Override
public void afterTextChanged(Editable arg0) {

```

Βάζουμε στο πεδίο εισαγωγής κειμένου (*EditText*) ένα *listener* αλλαγής κειμένου για να ελέγχουμε αν και πως μεταβάλλονται τα στοιχεία του πεδίου από τον χρήστη.

```

if
(! (editText_11.getText().toString().equals("")) && (arg0.charAt(0) != '0')) {

```

Βάζουμε έλεγχο στο πεδίο για να μην μπορεί ο χρήστης ο να αφήσει κενό το πεδίο αυτό και για να μην μπορεί να βάλει αριθμούς που αρχίζουμε με το ψηφίο μηδέν(0), όπως είναι το 001, 011, 000.

```

savePreferences("max_age", editText_11.getText()
.toString());
}
editText_11.setSelection(editText_11.length());

```

Σώζουμε το πεδίο στις προτιμήσεις του χρήστη και χρησιμοποιούμε τη φράση *setSelection* για να πάει ο *cursor* εισαγωγής στο τελευταίο ψηφίο.

```

}

@Override
public void beforeTextChanged(CharSequence s, int start,
int count,
int after) {

```

```

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int
before,
        int count) {
    }
    });
}

private void loadSavedPreferences() {
    String w = sharedPreferences.getString("max_weight", "200");
    String h = sharedPreferences.getString("max_height", "200");
    String a = sharedPreferences.getString("max_age", "100");
    editText_9.setText(w);
    editText_10.setText(h);
    editText_11.setText(a);
    editText_9.setSelection(editText_9.length());
    editText_10.setSelection(editText_10.length());
    editText_11.setSelection(editText_11.length());
}

```

Η *loadSavedPreferences()* καλείται στην αρχή της *Activity* μας . Καλείται δηλαδή όταν ανοίγει ο χρήστης την οθόνη των ρυθμίσεων . Έχουμε τρεις μεταβλητές που φορτώνουν τις προηγούμενες προτιμήσεις του χρήστη αν αυτές υπάρχουν , αν δεν υπάρχουν παίρνει την *default* τιμή που έχουμε εμείς ορίσει . Η *default* τιμή ουσιαστικά χρησιμοποιείται όταν μπαίνει ο χρήστης για πρώτη φορά στη οθόνη των ρυθμίσεων της εφαρμογής . Έπειτα παίρνουμε τις τιμές αυτές και τις ορίζουμε σαν τιμές των πεδίων που εμφανίζονται στο χρήστη και ορίζουμε τον *cursor* στο τέλος των τιμών αυτών , ώστε να μπορεί ο χρήστης να τις σβήσεις και να τις αλλάξει εύκολα.

```

private void savePreferences(String key, String value) {
    Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
}

```

```
        editor.commit();  
    }  
}
```

Η `savePreferences(String key, String value)` καλείται για να μας βοηθήσει να σώσουμε τις προτιμήσεις του χρήστη . Δέχεται σαν όρισμα το κωδικό όνομα της προτίμησης και την τιμή που έχει το πεδίο αντίστοιχο πεδίο. Όλα αυτά μετά από κάθε αλλαγή που κάνει στα πεδία ο χρήστης. Τέλος ένα αντικείμενο `Editor` χρησιμοποιείται για να υποβληθούν οι αλλαγές.

Πανεπιστήμιο Πειραιώς

MENU SCREEN

Find Diet

Settings

Help

Exit

Εικόνα 25: Menu Screen (3)

Settings Screen

Max Weight

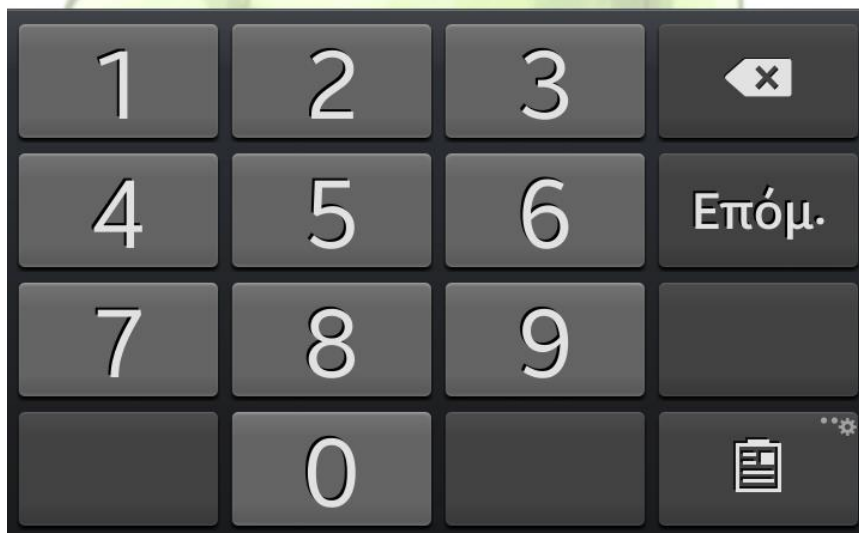
200

Max Height

200

Max Age

100



Εικόνα 26: Settings Screen

5.2.12 finddietlayout.xml

Η κλάση *FindDietActivity.java* είναι που σημαντική για την εφαρμογή μας. Αποτελεί μια από τις βασικότερες καθώς σε αυτή ο χρήστης προσθέτει τα στοιχεία που του ζητεί το πρόγραμμα, σχετικά με τη διαίτα και του παράγοντες της και βλέπει αμέσως τα αποτελέσματα των δεικτών (π.χ. δείκτης μάζας σώματος) που τον ενδιαφέρουν. Όταν έχουν συμπληρωθεί όλες οι παράμετροι ο χρήστης μπορεί να μεταβεί στην οθόνη των αποτελεσμάτων, δηλαδή της προτεινόμενης διαίτας. Έχει δοθεί μεγάλη έμφαση στο *interface* της οθόνης – δραστηριότητας αυτής, καθώς και στη βοήθεια με *hints* στα πεδία που πρέπει να εισάγει κάποια πληροφορία ο χρήστης. *Hints* υπάρχουν και στη καρτέλα των αποτελεσμάτων των δεικτών, για να επισημάνουν στο χρήστη ποια πληροφορία δεν έχει δώσει.

Η κλάση αυτή ενσωματώνει ένα βασικό πόρο διάταξης *layout* (*finddietlayout.xml*), που περιέχει και επεκτείνεται με *<include>* σε άλλους δυο (*input_data_layout.xml*, *calculator_data_layout.xml*). Τέλος στη κλάση αυτή χρησιμοποιούνται και άλλη τρεις μικρότερη πόροι διάταξης για μικρότερες ενέργειες, ο *spinner_item.xml* για να δείξει το πώς θα εμφανίζεται ένα αντικείμενο ενός *spinner* και *inputtab_spinner_borders.xml* και *inputtab_calculator_tab_edittext_borders.xml* για να προστεθούν κάποια περιγράμματα στους *spinners* και τα *edittexts* αντίστοιχα (διαφορετικά από τα *defaults*).

```
<?xml version="1.0" encoding="UTF-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imgV5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:contentDescription="@string/find_diet"
        android:src="@drawable/finddiet_title" />

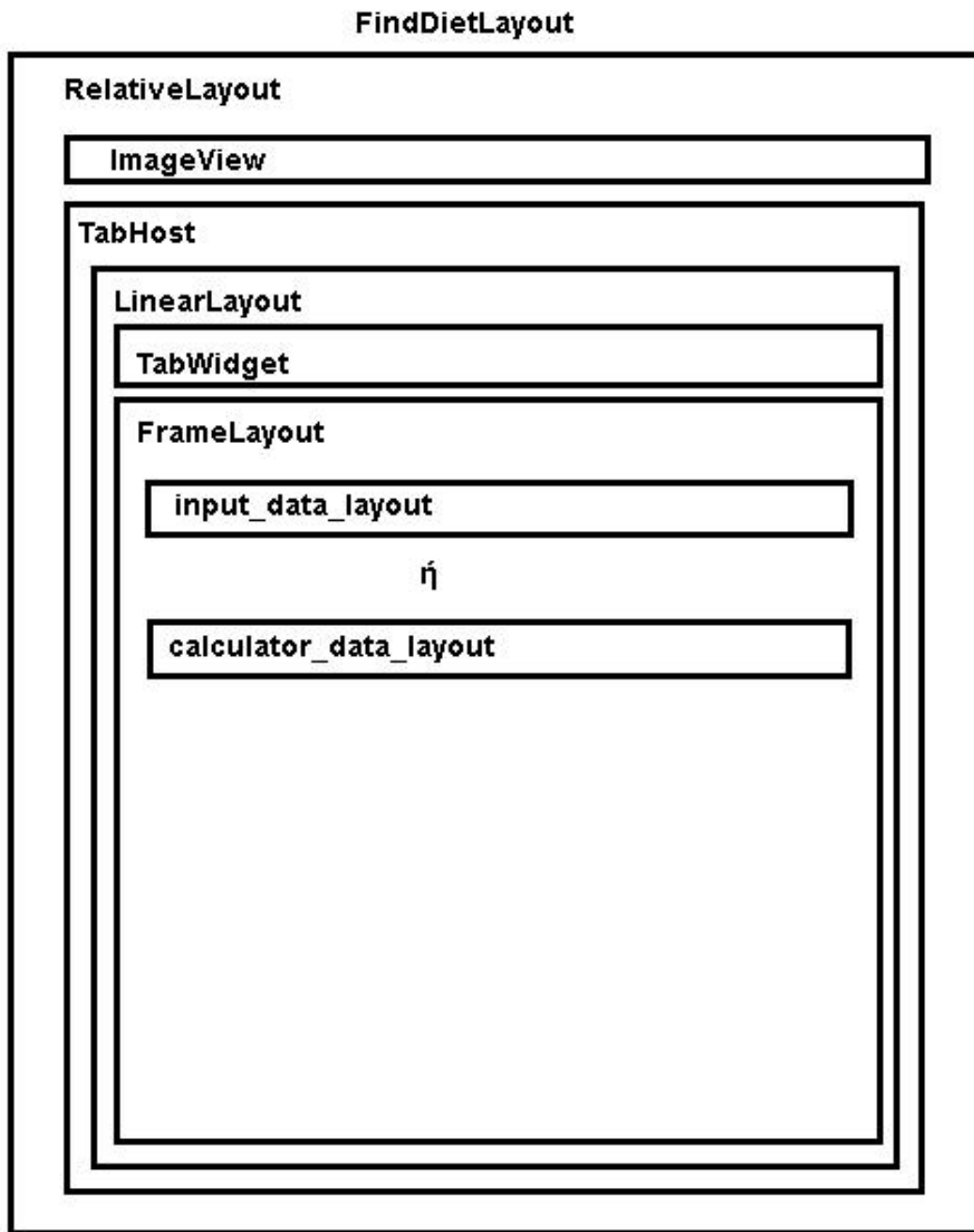
    <TabHost
        android:id="@+id/tabhost1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/imgV5" >

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

            <TabWidget
                android:id="@android:id/tabs"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content">
            </TabWidget>

            <FrameLayout
                android:id="@android:id/tabcontent"
                android:layout_width="fill_parent"
```

```
        android:layout_height="fill_parent" >
        <include
            android:id="@+id/include1"
            layout="@Layout/input_data_layout" >
        </include>
        <include
            android:id="@+id/include2"
            layout="@Layout/calculator_data_layout" >
        </include>
    </FrameLayout>
</LinearLayout>
</TabHost>
</RelativeLayout>
```



Εικόνα 27: finddietlayout.xml

Αξίζει να σημειωθεί ότι το παραπάνω *layout* ενσωματώνει άλλα δυο *layouts* από τα οποία ένα κάθε φορά εμφανίζεται στο χρήστη και επιλέγεται από αυτόν με την εναλλαγή των *tabs*. Τα δυο *layouts* αυτά ακολουθούν.

5.2.13 input_data_layout.xml

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

Ανάπτυξη Android Εφαρμογής «Δίαιτα (DietDroid)»


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >

    <ScrollView
        android:id="@+id/scroll1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true" >

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textV3"
                style="@style/textviews_inputtab"
                android:layout_alignParentTop="true"
                android:text="@string/baros" >
            </TextView>

            <EditText
                android:id="@+id/editT1"
                style="@style/edittexts_inputtab"
                android:layout_below="@+id/textV3"
                android:hint="@string/empty" >
            </EditText>

            <TextView
                android:id="@+id/textV4"
                style="@style/textviews_inputtab"
                android:layout_below="@+id/editT1"
                android:text="@string/yposos" >
            </TextView>

            <EditText
                android:id="@+id/editT2"
                style="@style/edittexts_inputtab"
                android:layout_below="@+id/textV4"
                android:hint="@string/empty" >
            </EditText>

            <TextView
                android:id="@+id/textV5"
                style="@style/textviews_inputtab"
                android:layout_below="@+id/editT2"
                android:text="@string/ilikia" >
            </TextView>

            <EditText
                android:id="@+id/editT3"
                style="@style/edittexts_inputtab"
```

```

        android:layout_below="@+id/textV5"
        android:hint="@string/empty" >
</EditText>

<TextView
    android:id="@+id/textV6"
    style="@style/textviews_inputtab"
    android:layout_below="@+id/editT3"
    android:text="@string/fyllo" >
</TextView>

<ImageSwitcher
    android:id="@+id/imsw1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textV6"
    android:layout_gravity="center"
    android:background="@color/inputtab_imageswitcher_background"
>

</ImageSwitcher>

<TextView
    android:id="@+id/textV7"
    style="@style/textviews_inputtab"
    android:layout_below="@+id/imsw1"
    android:text="@string/drastiriothta_entos_douleias" >
</TextView>

<LinearLayout
    android:id="@+id/lin1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textV7"
    android:background="@color/inputtab_linearlayout_background"
    android:orientation="vertical" >

    <ViewSwitcher
        android:id="@+id/viewswitcher1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:background="@color/inputtab_viewschwicher_background" >

        <TextView
            android:id="@+id/textV11"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

android:layout_gravity="center_horizontal|center_vertical"
            android:background="@layout/inputtab_spinner_borders"
            android:gravity="center"
            android:hint="@string/empty"
            android:orientation="vertical"

```

```

        android:textColorHint="@color/inputtab_spinner_hint"
        android:textSize="@dimen/inputtab_spinner_textsize" >
    </TextView>

    <Spinner
        android:id="@+id/spin1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/inputtab_spinner_background"
        android:spinnerMode="dialog" >
    </Spinner>
</ViewSwitcher>
</LinearLayout>

<TextView
    android:id="@+id/textV8"
    style="@style/textviews_inputtab"
    android:layout_below="@+id/Lin1"
    android:text="@string/drastiriothta_ektos_douleias" >
</TextView>

<LinearLayout
    android:id="@+id/Lin2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textV8"
    android:background="@color/inputtab_linearlayout_background"
    android:orientation="vertical" >

    <ViewSwitcher
        android:id="@+id/viewswitcher2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/inputtab_viewschwicher_background" >

        <TextView
            android:id="@+id/textV12"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal|center_vertical"
            android:background="@layout/inputtab_spinner_borders"
            android:gravity="center"
            android:hint="@string/empty"
            android:orientation="vertical"
            android:textColorHint="@color/inputtab_spinner_hint"
            android:textSize="@dimen/inputtab_spinner_textsize" >
        </TextView>

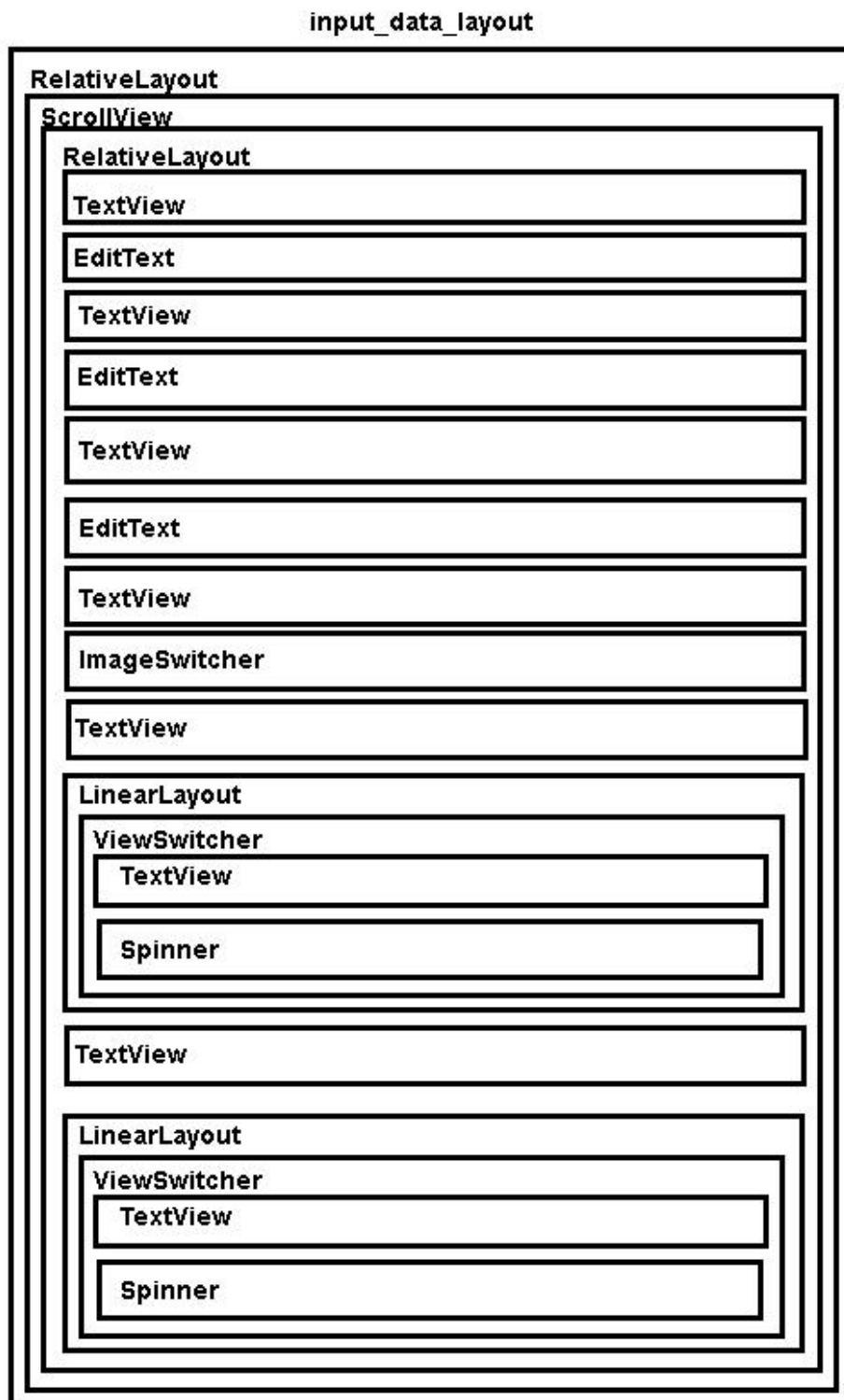
        <Spinner
            android:id="@+id/spin2"

```

```
                android:layout_width="match_parent"
                android:layout_height="wrap_content"

android:background="@color/inputtab_spinner_background"
                android:spinnerMode="dialog" >
                </Spinner>
            </ViewSwitcher>
        </LinearLayout>
    </RelativeLayout>
</ScrollView>
</RelativeLayout>
```

Πανεπιστήμιο Πειραιώς

**Εικόνα 28: input_data_layout.xml**

Αξίζει να σημειωθεί ότι σε αυτό το *layout* εκτός *TextViews* & *EditTexts* χρησιμοποιούνται *ImageSwitchers* , *ViewSwitchers* και *Spinners* . Χρησιμοποιούμε όλες αυτές τις διαφορετικές τεχνικές απεικόνισης του *android* για λόγους εκμάθησης , ποικιλίας και έγερση του ενδιαφέροντος του χρήστη για το *user interface*.

5.2.14 calculator_data_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >

    <ScrollView
        android:id="@+id/scroll2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true" >

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textV13"
                style="@style/textviews_calculatortab"
                android:layout_alignParentLeft="true"
                android:text="@string/deiktis_mazas_swmatos" >
            </TextView>

            <EditText
                android:id="@+id/editT4"
                style="@style/edittexts_calculatortab"
                android:layout_below="@+id/textV13" >
            </EditText>

            <TextView
                android:id="@+id/textV14"
                style="@style/textviews_calculatortab"
                android:layout_below="@+id/editT4"
                android:text="@string/deiktis_mazas_swmatos_kategoria" >
            </TextView>

            <EditText
                android:id="@+id/editT5"
                style="@style/edittexts_calculatortab"
                android:layout_below="@+id/textV14" >
            </EditText>

            <TextView
                android:id="@+id/textV15"
                style="@style/textviews_calculatortab"
```

```

        android:layout_below="@+id/editT5"
        android:text="@string/basikos_metabolismos" >
    </TextView>

    <EditText
        android:id="@+id/editT6"
        style="@style/edittexts_calculatortab"
        android:layout_below="@+id/textV15" >
    </EditText>

    <TextView
        android:id="@+id/textV16"
        style="@style/textviews_calculatortab"
        android:layout_below="@+id/editT6"
        android:text="@string/fysiki_drastiriotita" >
    </TextView>

    <EditText
        android:id="@+id/editT7"
        style="@style/edittexts_calculatortab"
        android:layout_below="@+id/textV16" >
    </EditText>

    <TextView
        android:id="@+id/textV17"
        style="@style/textviews_calculatortab"
        android:layout_below="@+id/editT7"
        android:text="@string/imerisies_thermidikes_anagkes" >
    </TextView>

    <EditText
        android:id="@+id/editT8"
        style="@style/edittexts_calculatortab"
        android:layout_below="@+id/textV17" >
    </EditText>

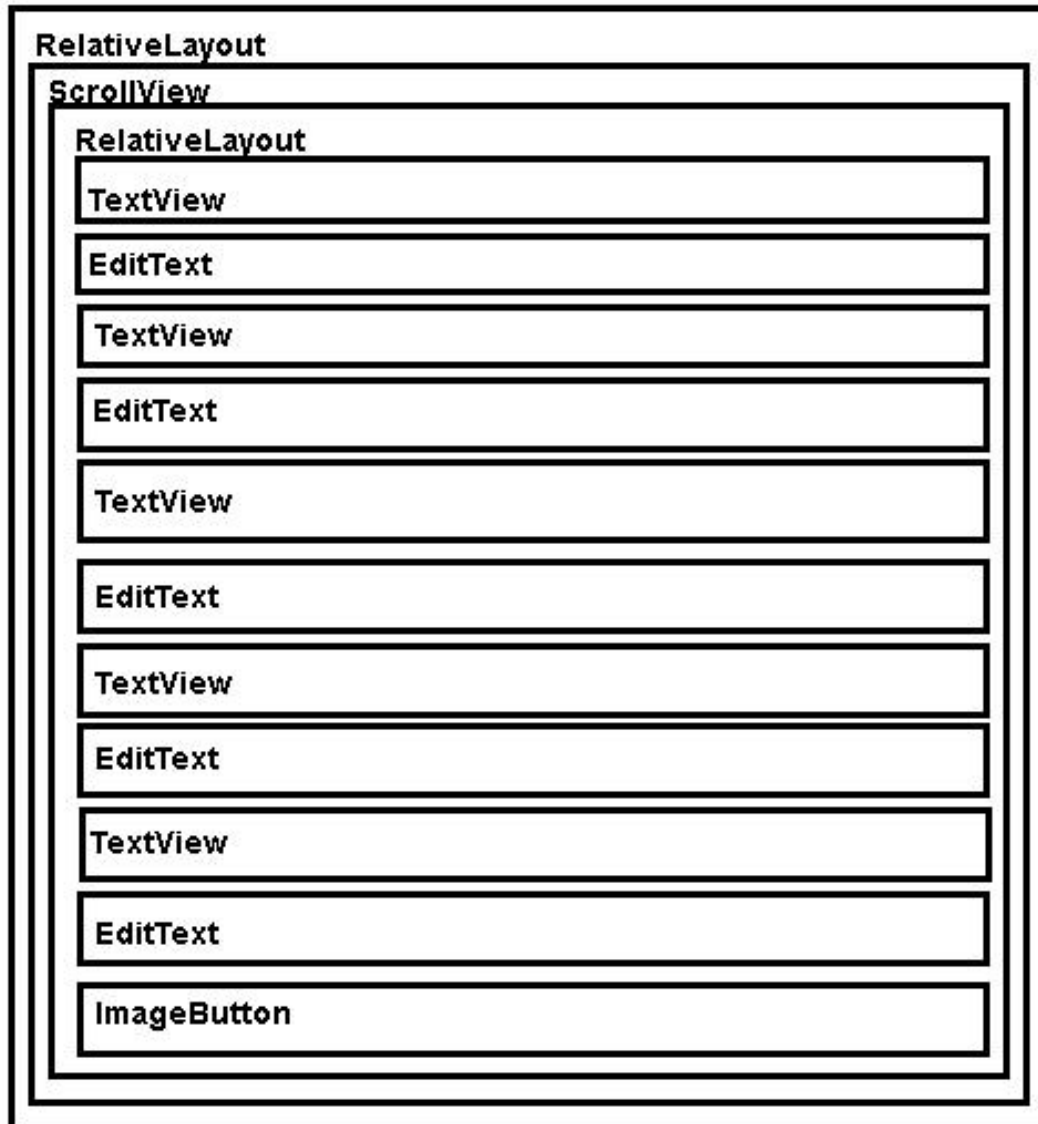
    <ImageButton
        android:id="@+id/imgbut1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editT8"
        android:layout_centerInParent="true"
        android:contentDescription="@string/find_diet_item"
        android:src="@drawable/icon_2"

        android:layout_marginTop="@dimen/calculatortab_imagebutton_margintop"
        android:background="@color/calculatortab_imagebutton"
        android:clickable="false">
    </ImageButton>
    </RelativeLayout>
</ScrollView>

</RelativeLayout>

```

calculator_data_layout



Εικόνα 29: `calculator_data_layout.xml`

Το `calculator_data_layout.xml` είναι ένας πόρος διάταξης που χρησιμοποιεί `TextViews` και `EditTexts` για να απεικονίσει τα αποτελέσματα των δεικτών του χρήστη άμεσα, με το που δίνει εκείνος πληροφορίες στο πρώτο tab. Το `ImageButton` που μας μεταφέρει στη οθόνη – δραστηριότητα της προτεινομένης δίαιτας, είναι κρυφό και εμφανίζεται μόνο όταν έχουν συμπληρωθεί όλα τα στοιχεία που ζητά η εφαρμογή από το χρήστη και κατά συνέπεια και όλοι η δείκτες της εφαρμογής.

5.2.15 spinner_item.xml

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textV10"
```



```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal|center_vertical"
    android:background="@Layout/inputtab_spinner_borders"
    android:gravity="center"
    android:hint="@string/empty"
    android:orientation="vertical"
    android:textColor="@color/inputtab_spinner_textcolor"
    android:textColorHint="@color/inputtab_spinner_hint"
    android:textSize="@dimen/inputtab_spinner_textsize" >

</TextView>

```

Ο παραπάνω κώδικας μας δίνει τη μορφοποίηση των *Spinners* που χρησιμοποιούνται στην *FindDietScreen*.

5.2.16 inputtab_spinner_borders.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <solid android:color="#00000000" >
    </solid>

    <stroke
        android:width="1dip"
        android:color="@color/inputtab_spinner_border" >
    </stroke>

</shape>

```

Ο παραπάνω κώδικας μας δίνει τη μορφοποίηση των περιγραμμάτων των *Spinners* που χρησιμοποιούνται στην *FindDietScreen*.

5.2.17 inputtab_calculatortab_edittext_borders.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <solid android:color="#00000000" >
    </solid>

    <stroke
        android:width="1dip"
        android:color="@color/inputtab_calculatortab_edittext_border" >
    </stroke>

</shape>

```

Ακολουθεί η παρουσίαση δυο αρχείων πόρων , που τοποθετούνται στο φάκελο των πόρων σχεδίασης και χρησιμοποιούνται στα *Tab* μας για να δηλώσουν τα χρώματα στις διαφορετικές καταστάσεις τους π.χ. όταν έχει πατηθεί το *Tab* .

5.2.18 tab_background_selectors.xml

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Active tab , selected -->
    <item android:drawable="@color/tab_color_selected"
        android:state_focused="false" android:state_pressed="false"
        android:state_selected="true"/>
    <!-- Inactive tab -->
    <item android:drawable="@color/tab_color_inactive"
        android:state_focused="false" android:state_pressed="false"
        android:state_selected="false"/>
    <!-- Pressed tab -->
    <item android:drawable="@color/tab_color_pressed"
        android:state_pressed="true"/>
</selector>
```

Βάζουμε χρώμα διαφανές για να υπάρχουν και οι αντίστοιχοι *selectors* στα γράμματα των *Tabs* και στο *background* των *Tabs*. Αν δεν βάζαμε διαφανές χρώμα ο *background selectors* καλύπτει τα άλλους .

5.2.19 tab_text_selectors.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true"
        android:color="@color/tab_textcolor_pressed"/> <!-- pressed -->
    <item android:state_focused="true"
        android:color="@color/tab_textcolor_focused"/> <!-- focused -->
    <item android:color="@color/tab_textcolor_default"/> <!-- default -->
</selector>
```

Ακολουθεί η ανάλυση της οθόνης – δραστηριότητας *FindActivity.java* .

5.2.20 FindDietActivity.java

Η *FindDierActivity* βρίσκεται μέσα στο πακέτο *com.example.dietdroid* . Χρησιμοποιεί ένα αντικείμενο *FullScreen* για να καλύπτει ο πόρος *layout* που φορτώνεται (*finddietlayout.xml*) ολόκληρη την οθόνη του κινητού. Φορτώνονται επίσης κατά το άνοιγμα τις οθόνης και οι προτιμήσεις που έχει δηλώσει ο χρήστης , για περιορισμούς στα παρακάτω *EditTexts*. Ακολουθεί ο κώδικας για το *TabHost* .

Γίνεται *setup* το *TabHost* για να αρχικοποιηθεί. Ακολουθεί ο κώδικας για το πρώτο *Tab* . Ορίζουμε τίτλο και *icon* , ορίζω ποιος πόρος *layout* θα χρησιμοποιηθεί όταν είναι ανοιχτό το συγκεκριμένο *Tab* (*include1=>input_data_layout.xml*) και το προσθέτουμε στο *TabHost* μας . Ακολουθεί ο κώδικας για το δεύτερο *Tab* . Στο δεύτερο *Tab* μας , ορίζουμε τίτλο και *icon* , ορίζουμε

ποιος πόρος *layout* θα χρησιμοποιηθεί όταν είναι ανοιχτό το συγκεκριμένο *Tab* (*include2=>calculator_data_layout.xml*) και το προσθέτουμε στο *TabHost* μας .

Στην εφαρμογή μας επιθυμούμε να βάλουμε στα *Tabs* και *icon* και *text*. Όμως επειδή η εφαρμογή κτίζεται και για κινητά τηλέφωνα με λειτουργικό έως android 4+ υπάρχει πρόβλημα . Δεν προβλέπεται να τηλέφωνα με λειτουργικό από android 4+ η χρήση και *icon* και *text* ταυτόχρονα με την εντολή *android.widget.TabHost.TabSpec.setIndicator(CharSequence label, Drawable icon)* . Για αυτό χρησιμοποιούμε τις *LayoutParameters* του *TabWidget* για να μεγαλώσουμε το ύψος των *Tabs* και να χωρέσουν και *texts* και *icons* . Έπειτα προσθέτουμε τις χρωματικές επιλογές για τις διάφορες καταστάσεις των *Tabs* για το *background* και τις λέξεις και προσθέτουμε και *strip* (λωρίδα επιλογής για τα *Tabs*). Τέλος ορίζουμε ποιο θα είναι το αρχικά ορατό *Tab* μας.

Κρύβουμε αρχικά το *ImageButton* που βρίσκεται στο δεύτερο *Tab* μας και θα το εμφανίσουμε μόνο αν έχουν υπάρξει κάποιες προϋποθέσεις στη συνέχεια . Χρησιμοποιούμε ένα *Listener* για την εναλλαγή μας στα *Tabs* . Συμβάντα στην εφαρμογή μας συμβαίνουν όταν πάμε από το πρώτο στο δεύτερο *Tab* . Βάζουμε μια *if* ερώτηση για να δούμε αν βρισκόμαστε στο δεύτερο *Tab* και έπειτα η πρώτη κίνηση μας είναι να κρύψουμε το πληκτρολόγιο , σε περίπτωση που έχει μείνει ανοικτό από το προάγουμε *Tab* , για να μην μας κρύβει την οθόνη του δευτέρου (είναι μια ενέργεια που δεν γίνεται αυτόματα από το λειτουργικό) . Χρησιμοποιούμε το *editT_4* γιατί είναι το πρώτο *EditText* στο δεύτερο *Tab* .

Έπειτα θέλουμε σαν συμβάν να εμφανίζεται ένα μικρό *Toast* μήνυμα που να δείχνει τις τιμές που έβαλε ο χρήστης στο πρώτο *Tab* (αν έχει βάλει) και να είναι μικρής διάρκειας.

Έπειτα ένα μεγάλο *block* κώδικα ουσιαστικά εφαρμόζει τους υπολογισμούς που προκύπτουν από τους τύπους του διαιτολόγου για να υπολογιστεί το *BMI* και η κατηγορία που ανήκει ο χρήστης , το *BMR* , ο παράγοντας ασκήσεις και οι ημερήσιες θερμίδες που πρέπει να καταναλώνει ο χρήστης. Επιπλέον στο παραπάνω κώδικα χρησιμοποιούνται πολλές ερωτήσεις *if* για να μπορούμε να δίνουμε στο χρήστη τα κατάλληλα μηνύματα για τα κενά *EditText* που δεν έχει συμπληρώσει . Ένα *icon* που μας μεταφέρει στη επόμενη οθόνη της εφαρμογής μας . Γράφεται εδώ γιατί μόνο όταν έχουμε συμπληρώσει όλα τα πεδία μας βγάζει αποτέλεσμα το *editT_8* , άρα μόνο τότε μπορούμε να μεταβούμε στη επόμενη οθόνη αφού είναι προϋπόθεση ορισμού της διαίτας οι ημερήσιες θερμιδικές ανάγκες. Χρησιμοποιούμε ακόμα κώδικα για να κρύψουμε ξανά το *icon* μετάβασης στην επόμενη οθόνη , όταν σβηστεί από το χρήστη κάποιο στοιχείο από αυτά που του ζητάμε . Στα *EditTexts* έχουμε βάλει *hints* για να δείχνουν στο χρήστη ότι δεν έχουν συμπληρωθεί. Κρατάμε σε μια μεταβλητή την τιμή του πεδίου κάθε φορά και έχουμε βαλε ένα *Listener* αλλαγής κειμένου . Οπότε αν ο χρήστης πάει να κάνει οποιαδήποτε αλλαγή στα *EditTexts* αυτή να καταγράφεται . Έχουμε βάλει και 2 περιορισμούς . Πρώτος περιορισμός είναι ο χρήστης να μην μπορεί να βάλει στο πεδίο τιμή μεγαλύτερη από την αντίστοιχη *max* (μέγιστη) που έχει δηλώσει στα *Settings*. Δεύτερος περιορισμός είναι ο χρήστης (επειδή του εμφανίζεται *numeric* πληκτρολόγιο για να συμπληρώσει το πεδίο) να μην μπορεί να χρησιμοποιήσει το μηδέν σαν πρώτο ψηφίο συμπλήρωσης του πεδίου . Έτσι αριθμοί τύπου αριθμοί τύπου 000 , 011 , 001 αποκλείονται αφού δεν είναι και λογικό να υπάρχουν. Τέλος μετά από οποιαδήποτε αλλαγή στο πεδίο ο *cursor* πάει στο τέλος ώστε να μπορούμε να το κουμπί *erase* του πληκτρολογίου να σβήσουμε κατευθείαν ότι δεν μας χρησιμεύει.

Στην εφαρμογή μας χρησιμοποιείται ένας *ImageSlider* στην οθόνη *input_data_layout.xml*. Επειδή θέλουμε σε όλα τα πεδία συμπληρώσεις που υλοποιούμε να βάλουμε ένα *hint* για να βοηθά το χρήστη και στον *ImageSlider* δεν προβλέπεται κάτι τέτοιο , προσθέσαμε μια επιπλέον εικόνα στη σειρά των εικόνων που φορτώνει (ένα ερωτηματικό) και με το πρώτο άγγιγμα του χρήστη αλλάζει και η εικόνα που εκείνος βλέπει , αλλά ουσιαστικά αφαιρείται και η εικόνα του *hint* από τις εικόνες που θα δείχνει ο *ImageSlider* . Οι εικόνες αλλάζουν με ένα *motion – action_down event* και προσδιορίζεται ανάλογα με το σε ποια εικόνα είμαστε αν είναι άνδρας ή γυναίκα ο χρήστης. Παρακατω ακολουθεί ο κώδικας για τους *Spinners*.

Στην εφαρμογή μας χρησιμοποιούνται *Spinners* στην οθόνη *input_data_layout.xml*. Στους

Spinners θέλαμε να προσθέσουμε *hint* , πράγμα που δεν προβλέπεται από το *android*. Έτσι χρησιμοποιούμε ένα *ViewSwitcher* , ώστε να κάνει την εναλλαγή από το ένα *EditText* που έχει *hint* σε ένα *Spinner*. Με ένα *Event (onClick)* γίνεται η εναλλαγή και ο χρήστης νομίζει ότι ο *Spinner* έχει το *hint* . Επίσης με το *Event* ανοίγει και ο *Spinner* , σαν να το έχεις κλικάρει στην αλήθεια. Δεν ξαναλειτουργεί ο *ViewSwitcher* και δεν ξαναγίνεται εναλλαγή σε *EditText*.

Η *makeView()* χρησιμοποιείται στον *ImageSlider* . Για να χρησιμοποιήσουμε ένα *ImageSlider* συνεργαζόμαστε με ένα *ViewFactory* και η απαιτούμε από το *ViewFactory* μέθοδος προς υλοποίηση είναι η *makeView()* . Δεν σημαίνει ότι η *makeView()* χρησιμοποιείται για καθορίσει το περιεχόμενο της *View* μας , αλλά χρησιμοποιείται σαν αρχικό *template* για το πώς μπορεί να εμφανίζει η *View* τα «παιδιά» της .

Η *calculate_male_bmr* και η *calculate_female_bmr* χρησιμοποιούνται για να υπολογίσουν το *BMR* για τους άνδρες και τις γυναίκες και να επιστρέψουν στη ροή την τιμή του . Ο υπολογισμός γίνεται με τους τύπους που έχουν δοθεί από το διαιτολόγο.

Πανεπιστήμιο Πειραιώς

MENU SCREEN

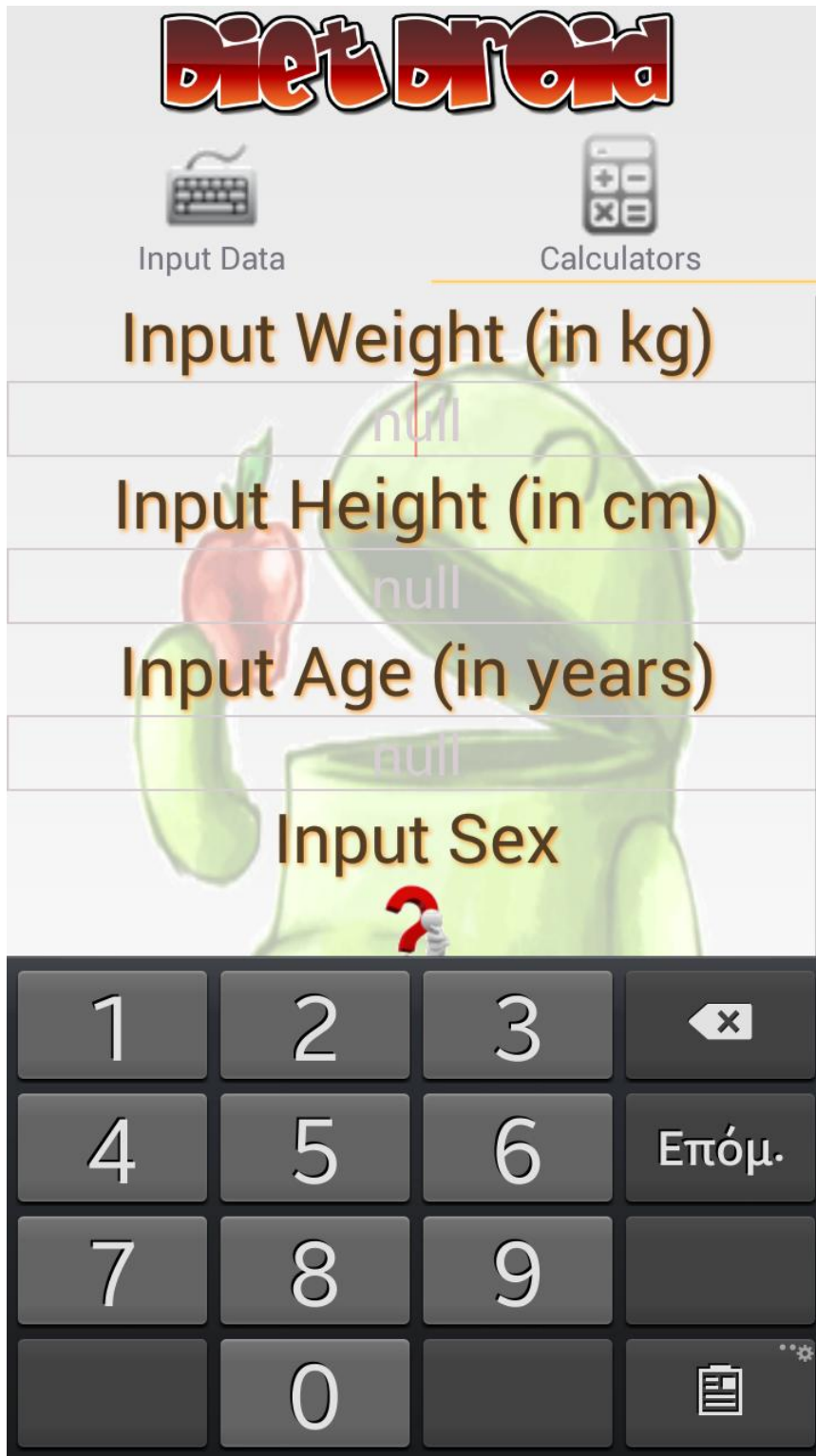
Find Diet

Settings

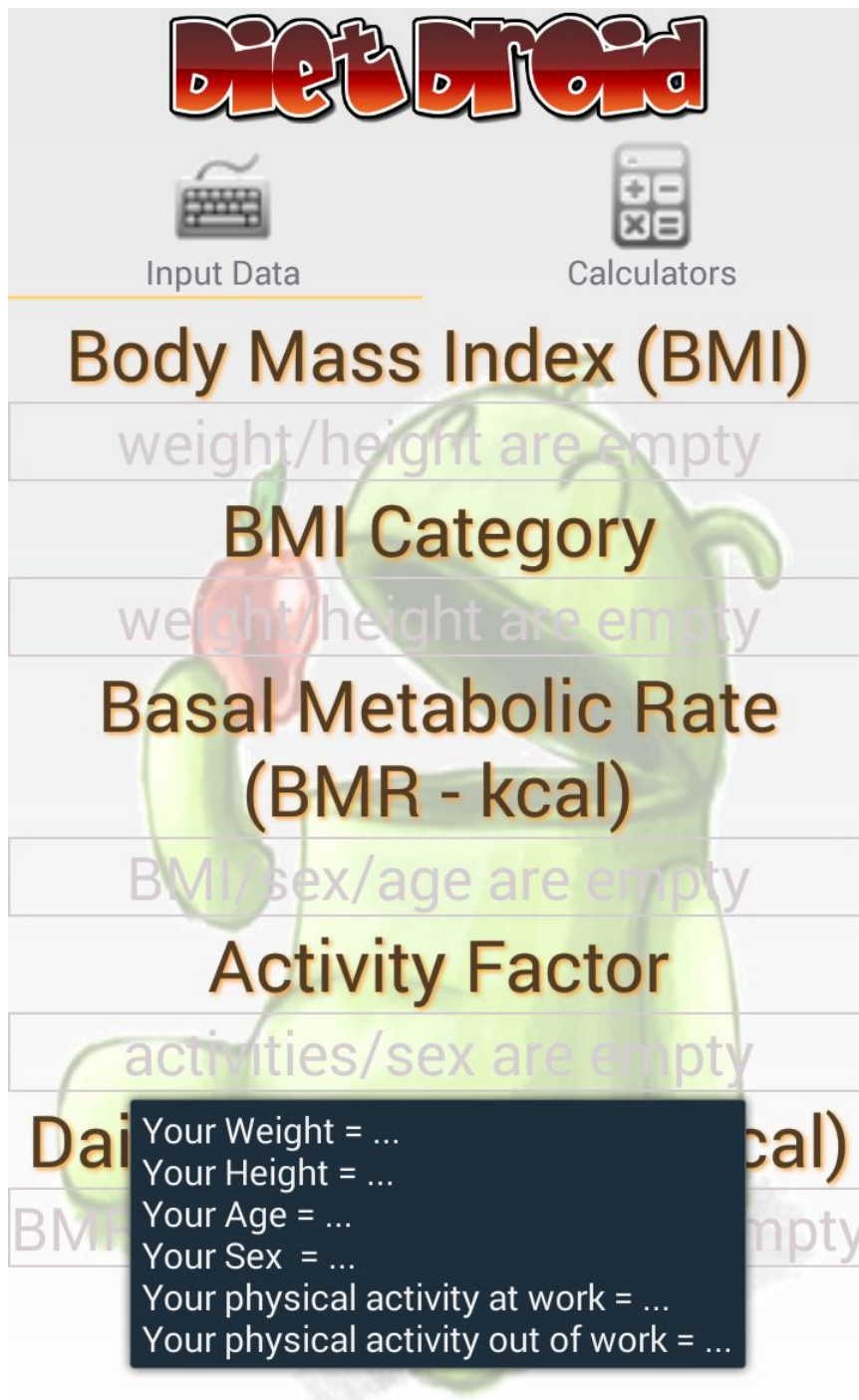
Help

Exit

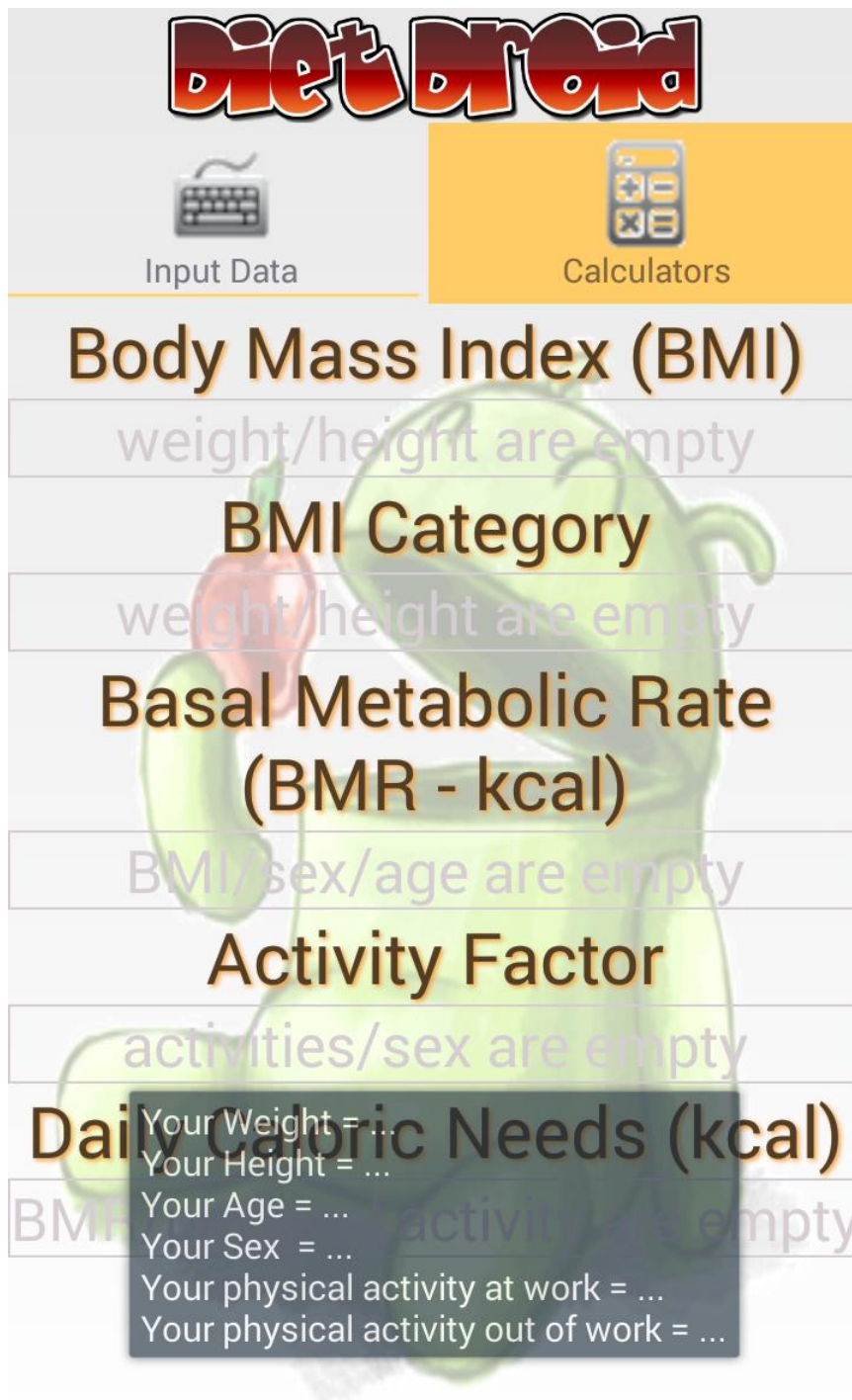
Εικόνα 30: Menu Screen (4)



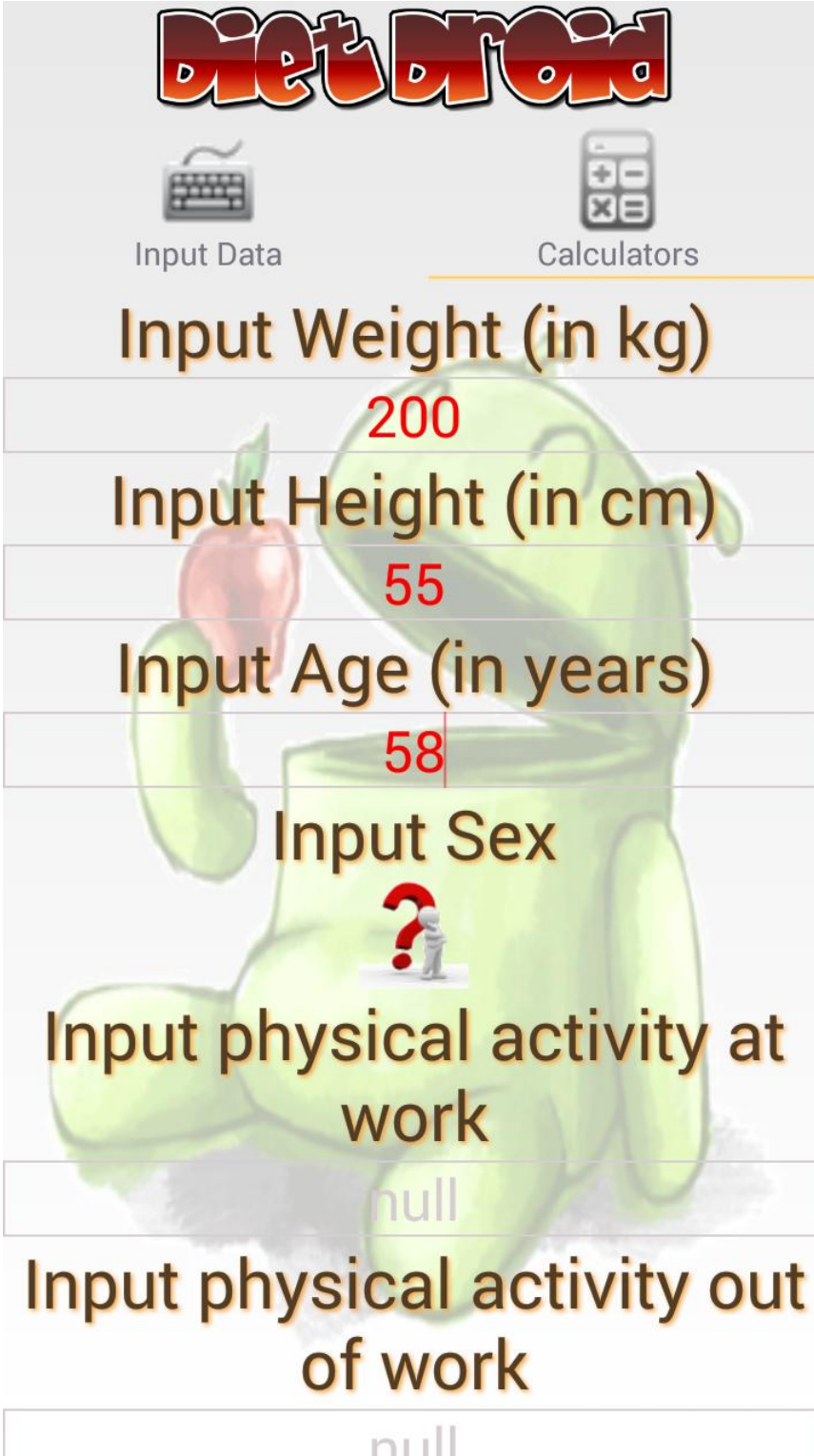
Εικόνα 31: FindDiet Screen (1)



Εικόνα 32: FindDiet Screen (2)



Εικόνα 33: FindDiet Screen (3)



The screenshot shows the 'Diet Droid' app interface. At the top, there is a title 'Diet Droid' in a stylized font. Below the title are two tabs: 'Input Data' and 'Calculators'. The 'Input Data' tab is selected. The form contains the following fields:

- Input Weight (in kg): 200
- Input Height (in cm): 55
- Input Age (in years): 58
- Input Sex: A red question mark icon with a small white figure standing next to it.
- Input physical activity at work: null
- Input physical activity out of work: null

Εικόνα 34: FindDiet Screen (4)

Diet Droid

Input Data Calculators

Input Weight (in kg)
200

Input Height (in cm)
55

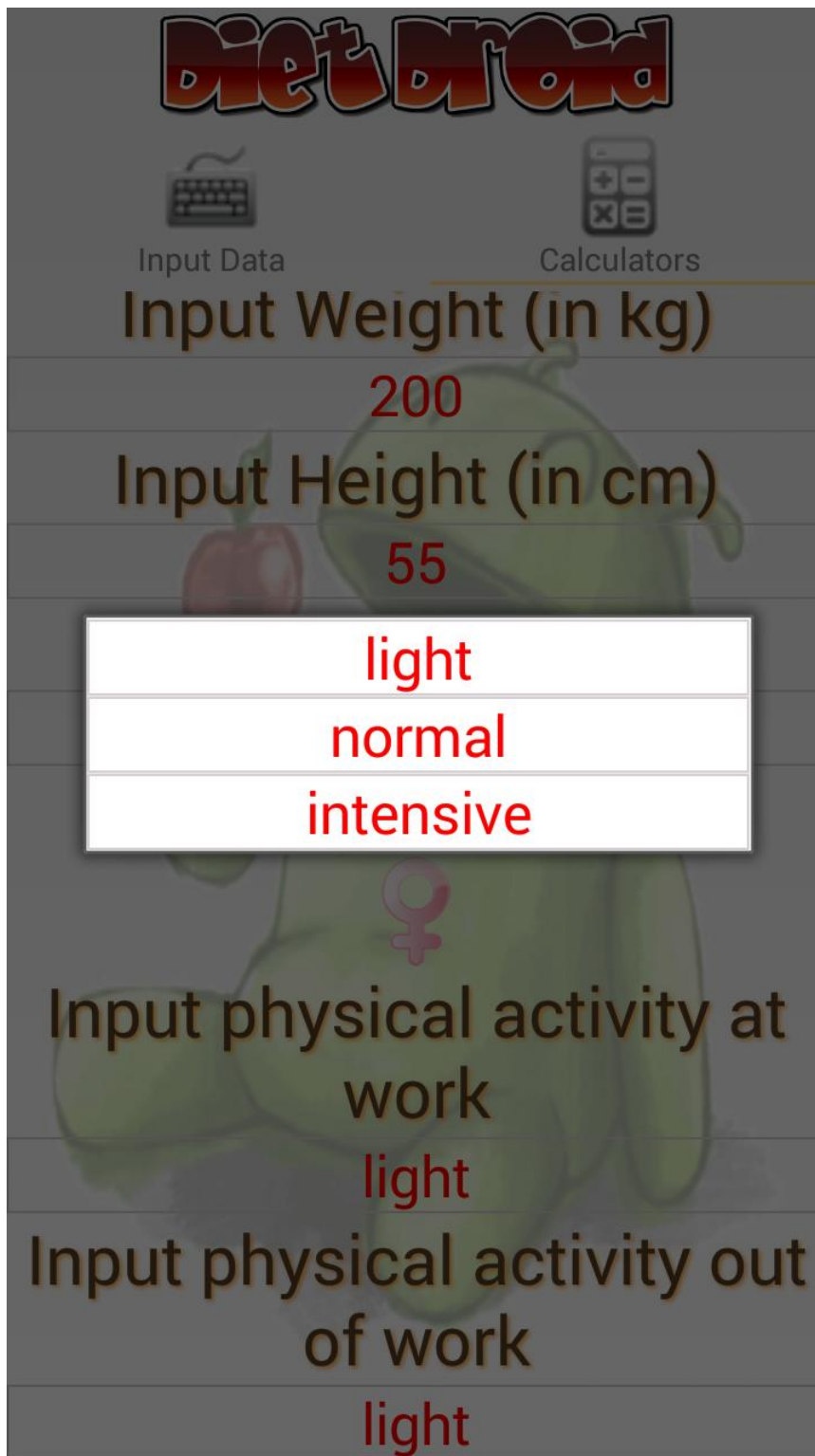
Input Age (in years)
58

Input Sex
♀ ♂

Input physical activity at work
null

Input physical activity out of work
null

Εικόνα 35: FindDiet Screen (5)



Εικόνα 36: FindDiet Screen (6)

5.2.21 resultslayout.xml

Είναι η οθόνη – δραστηριότητα που μας δείχνει τα αποτελέσματα της εφαρμογής μας. Μας δείχνει μια προτεινόμενη διαίτα για κάθε μέρα , ένα εβδομαδιαίο πρόγραμμα . Ο χρήστης μπορεί αν δεν ικανοποιηθεί από το πρόγραμμα μιας ημέρας με μια εναλλαγή (στις μέρες) να δει ένα διαφορετικό ημερήσιο πρόγραμμα για την συγκεκριμένα μέρα . Έχει δημιουργηθεί μια διαδικασία *random* επιλογής μεταξύ των επιτρεπτών τροφίμων (επιτρεπτόν σύμφωνα με τις μετρήσεις και τις τιμές που έχει βάλει ο χρήστης) και έτσι και κάθε μέρα βγάζει διαφορετικό πρόγραμμα σε σχέση με τις άλλες μέρες και αν δεν μας αρέσει μια μέρα (και οι άλλες μας αρέσουν) , μπορούμε να αλλάξουμε μόνο εκείνη τη μέρα με την διαδικασία που προαναφέραμε. Τέλος έχει δοθεί η επιλογή στο χρήστη να σώζει τα ημερήσια προγράμματα αυτά απευθείας στο *calendar* του , με έτοιμη την περιγραφή της διαίτας και τη μέρα , και μόνη υποχρέωση του χρήστη να ορίζει την ώρα της υπενθύμισης.

```
<?xml version="1.0" encoding="UTF-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imgV6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:contentDescription="@string/find_diet"
        android:src="@drawable/finddiet_title" />

    <TabHost
        android:id="@+id/tabhost2"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/imgV6" >

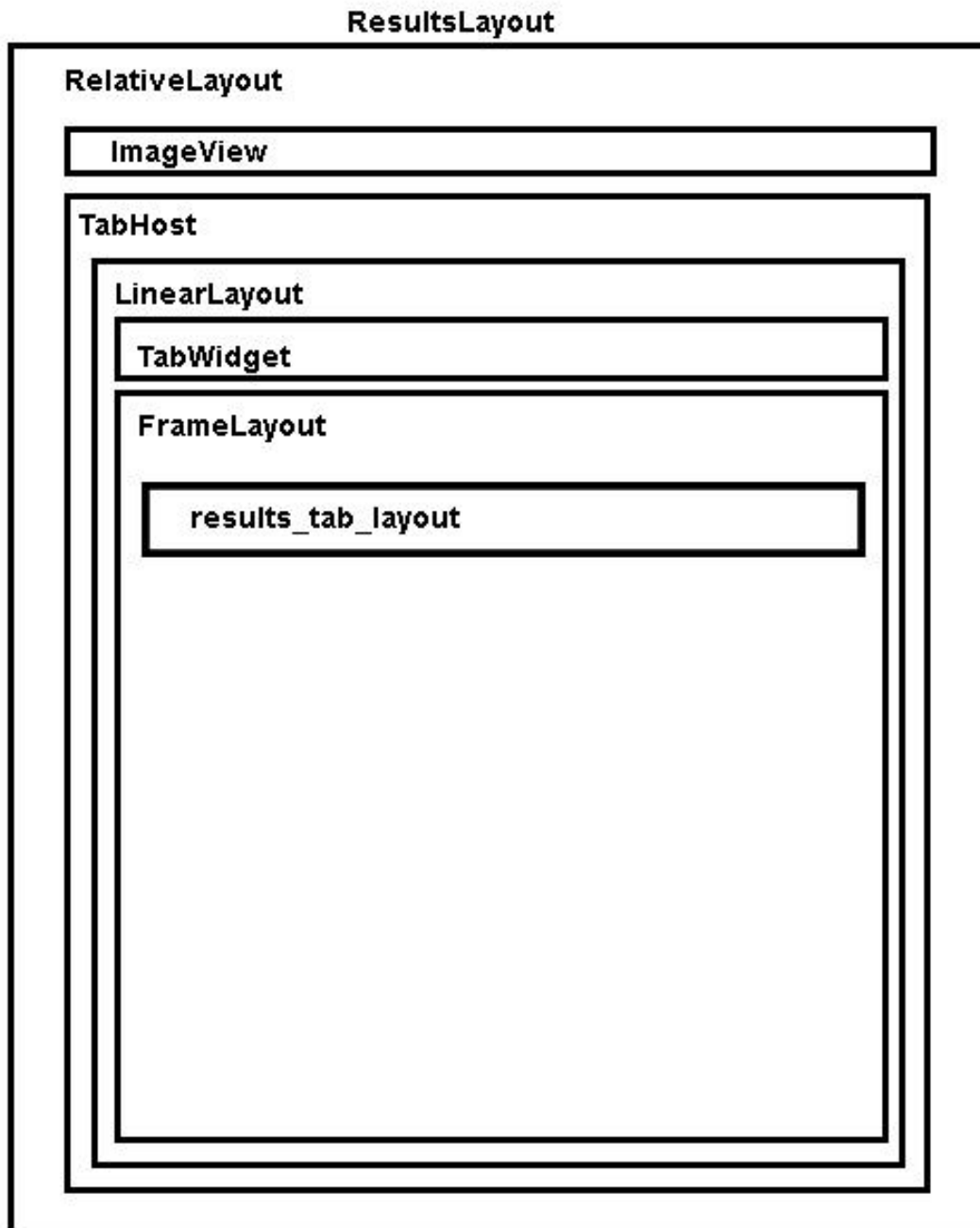
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:orientation="vertical" >

            <TabWidget
                android:id="@android:id/tabs"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content" >
            </TabWidget>

            <FrameLayout
                android:id="@android:id/tabcontent"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" >

                <include
                    android:id="@+id/include3"
                    layout="@layout/results_tab_layout" >
```

```
        </include>  
    </FrameLayout>  
    </LinearLayout>  
    </TabHost>  
</RelativeLayout>
```



Εικόνα 37: resultslayout.xml

5.2.22 results_tab_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background" >

    <ScrollView
        android:id="@+id/scroll4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true" >

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <TextView
                android:id="@+id/textV24"
                style="@style/text_type_1_results"
                android:layout_alignParentTop="true" >
            </TextView>

            <TextView
                android:id="@+id/textV25"
                style="@style/text_type_2_results"
                android:layout_below="@+id/textV24" >
            </TextView>

            <TextView
                android:id="@+id/textV26"
                style="@style/text_type_1_results"
                android:layout_below="@+id/textV25" >
            </TextView>

            <TextView
                android:id="@+id/textV27"
                style="@style/text_type_2_results"
                android:layout_below="@+id/textV26" >
            </TextView>

            <TextView
                android:id="@+id/textV28"
                style="@style/text_type_1_results"
                android:layout_below="@+id/textV27" >
            </TextView>

            <TextView
                android:id="@+id/textV29"
                style="@style/text_type_2_results"
                android:layout_below="@+id/textV28" >
            </TextView>
        </RelativeLayout>
    </ScrollView>
</RelativeLayout>
```

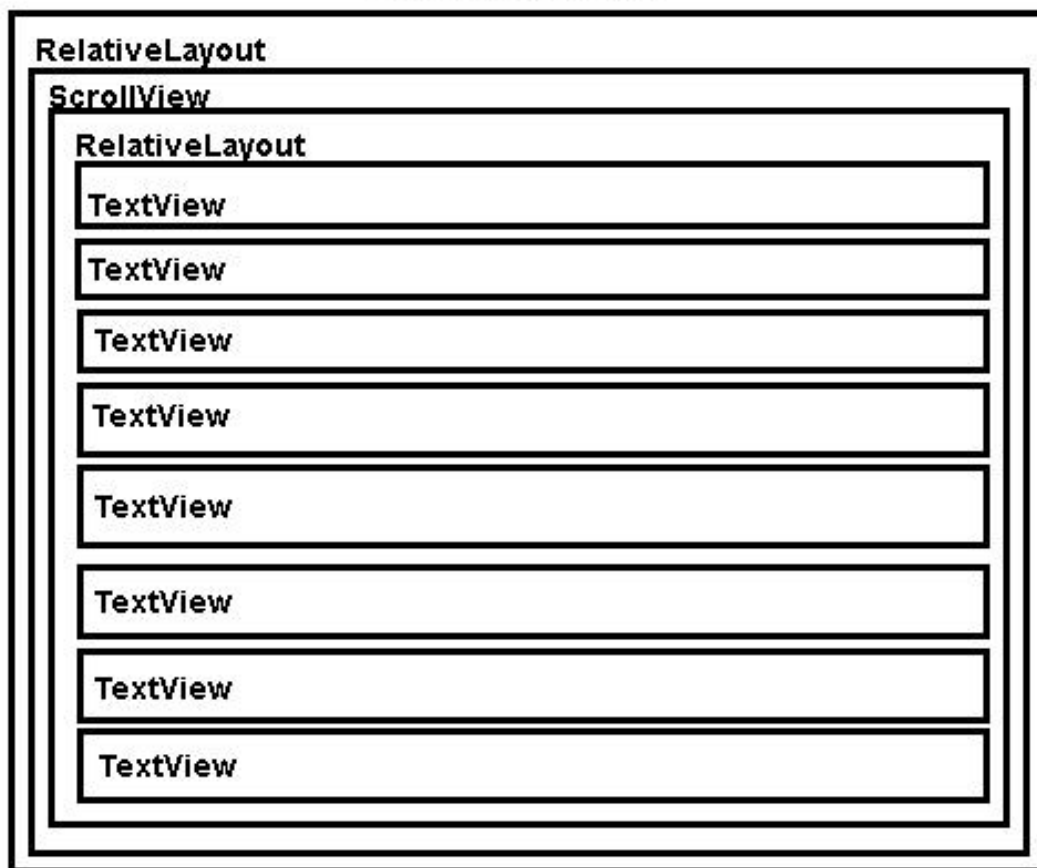
```

        <TextView
            android:id="@+id/textV30"
            style="@style/text_type_1_results"
            android:layout_below="@+id/textV29" >
        </TextView>

        <TextView
            android:id="@+id/textV31"
            style="@style/text_type_2_results"
            android:layout_below="@+id/textV30" >
        </TextView>
    </RelativeLayout>
</ScrollView>
</RelativeLayout>

```

results_tab_layout



Εικόνα 38: results_tab_layout.xml

Τα δυο παραπάνω αρχεία χρησιμοποιούνται για το *interface* της οθόνης - δραστηριότητας με τα αποτελέσματα της εφαρμογής μας . Παρακατω είναι ο κώδικας λειτουργίας της οθόνης - δραστηριότητας αυτής .

5.2.23 ResultsActivity.java

Η κλάση *ResultsActivity.java* ανήκει κι αυτή στο πακέτο *com.example.dietdroid* . Όταν καλείται , δημιουργείται ένα αντικείμενο *FullScreen* για να καλύπτεται ολόκληρη η οθόνη του κινητού από την εφαρμογή. Ορίζεται σαν layout ο πόρος διάταξης *resultslayout.xml* .

Έπειτα αρχικοποιούνται τα *TextViews* των τίτλων και σε παρένθεση είναι οι μέγιστες επιτρεπτές τιμές των πρωινών , μεσημεριανών και βραδινών αντίστοιχα . Επιπλέον ανοίγει και η επικοινωνία με τη βάση για να πάρουμε τις λίστες με τα πρωινά , μεσημεριανά και βραδινά. Κάνουμε *setup* στο *TabHost* . Δημιουργούμε τα 7 *tabs* μας , προσθέτουμε αναφορές σε αυτά , τίτλους , ορίζουμε με ποιο *layout* θα εμφανίζονται τα δεδομένα μας και τέλος προσθέτουμε όλα τα *tabs* στο *TabHost* . Μετά χρησιμοποιούμε κώδικα για να δηλώσουμε τα διαφορετικά *states* (χρωματικές καταστάσεις γραμμμάτων και *backgrounds*) των *tabs*. Ορίζονται τα *strips* που χρησιμοποιούνται για *tabs*.

Ορίζουμε το πρώτο *tab* (την Δευτέρα) σαν αρχικό – τωρινό *tab* . Στο κώδικα χρησιμοποιείται αρχικά σαν αρχικό το δεύτερο *tab* και έπειτα το πρώτο για ένα και μόνο λόγο , γιατί υπάρχει ένα *bug* του *android* στο κώδικα και δεν εμφανίζει σωστά (αλλά κενό) το πρώτο *tab* , αν πρώτα δεν μεταβούμε στο δεύτερο και ακαριαία στο πρώτο .

Ορίζουμε την τιμή που παίρνει το *TextView* αν η λίστα με τα πρωινά μετά τους υπολογισμούς του διαιτολόγου και την διαλογή είναι άδεια , δηλαδή αν δεν πληροί κανένα πρωινό τις προδιαγραφές για να προταθεί στο χρήστη. Ορίζουμε την τιμή που παίρνει το *TextView* αν η λίστα με τα μεσημεριανά μετά τους υπολογισμούς του διαιτολόγου και την διαλογή είναι άδεια , δηλαδή αν δεν πληροί κανένα μεσημεριανό τις προδιαγραφές για να προταθεί στο χρήστη.

Ορίζουμε τις τροφές που μπορεί να φάει ο χρήστης σε κάθε κατηγορία, όταν οι λίστες με πρωινά , μεσημεριανά και βραδινά δεν είναι άδειες. Η επιλογή των τροφών γίνεται με τυχαίο τρόπο από τις λίστες που προκύπτουν από την βάση. Όλα αυτά γίνονται για να γίνει η αρχικοποίηση του πρώτου *tab* και να δει ο χρήστης άμεσα αποτελέσματα . Πιο κάτω με την κάθε εναλλαγή ανάμεσα στα *tab* γίνεται η ίδια διαδικασία για το κάθε *tab*.

Έπειτα ακολουθεί ο κώδικας που χρησιμοποιείται για τις εναλλαγές ανάμεσα στα *tabs* . Έχουμε βάλει ένα *listener* για να μπορεί η εφαρμογή να αντιλαμβάνεται την εναλλαγή . Τώρα σε κάθε εναλλαγή από τις λίστες με τα επιτρεπτά πρωινά , μεσημεριανά , βραδινά επιλέγονται με μια *random* συνάρτηση κάποια από αυτά (1 από το καθένα) και απεικονίζονται στο χρήστη . Τέλος ο χρήστης μπορεί και να δει πιο είναι το όριο των θερμίδων που μπορεί να έχει σε ένα γεύμα , πόσες τελικά έχει το γεύμα του προτάθηκε αλλά και πόσες συνολικά θερμίδες θα πάρει την μέρα αν τελικά τηρήσει αυτό το ημερήσιο πρόγραμμα.

Έπειτα ακολουθεί ο κώδικας που χρησιμοποιείται για να εισάγουμε ένα ημερήσιο πρόγραμμα δίαιτας στο *calendar* και για να μας παρέχει κάποιες ευκολίες στο ορισμό υπενθύμισης για αυτό . Κάθε ένα *tab* ορίζεται σαν ξεχωριστό *View* για να μπορούμε να του προσθέσουμε ένα *onLongClick listener* . Μετά γραφούμε τη πρόθεση μας να κάνουμε μια ενέργεια εισαγωγής και δημιουργούμε το *Event* . Ορίζουμε την περιγραφή του *Event* (χρησιμοποιώντας το όνομα της κάθε ημέρας και τα στοιχεία της προτεινόμενης δίαιτας) . Έπειτα παίρνουμε ένα *instance* της του *calendar* , δηλαδή τι μέρα , μήνα , έτος έχουμε τώρα . Ανάλογα με το *tab* που είμαστε γνωρίζουμε την ημέρα που θέλουμε να καταχωρηθεί η δίαιτα , οπότε αν το όνομα αυτής δεν συμπίπτει με της σημερινής λέμε στο *calendar* να προσχωρήσει μια ημέρα εμπρός μέχρι να την βρει. Τέλος ορίζουμε το *event* σαν ολόημερο με αρχή και τέλος χρήσης την μέρα που ορίζεται.

Diet Droid

Monday Tuesd.. Wedne.. Thurs.. Friday Saturd.. Sunday

Breakfast (<=363)

Πιτάκια με μαρμελάδα (1 μικρό)

Lunch (<=605)

Κοτόσουπα με ρύζι αυγολέμονο
(1 μερίδα 350γρ.)

Dinner (<=242)

Ζαμπόν διαίτης (100 γρ.)

Total Calories

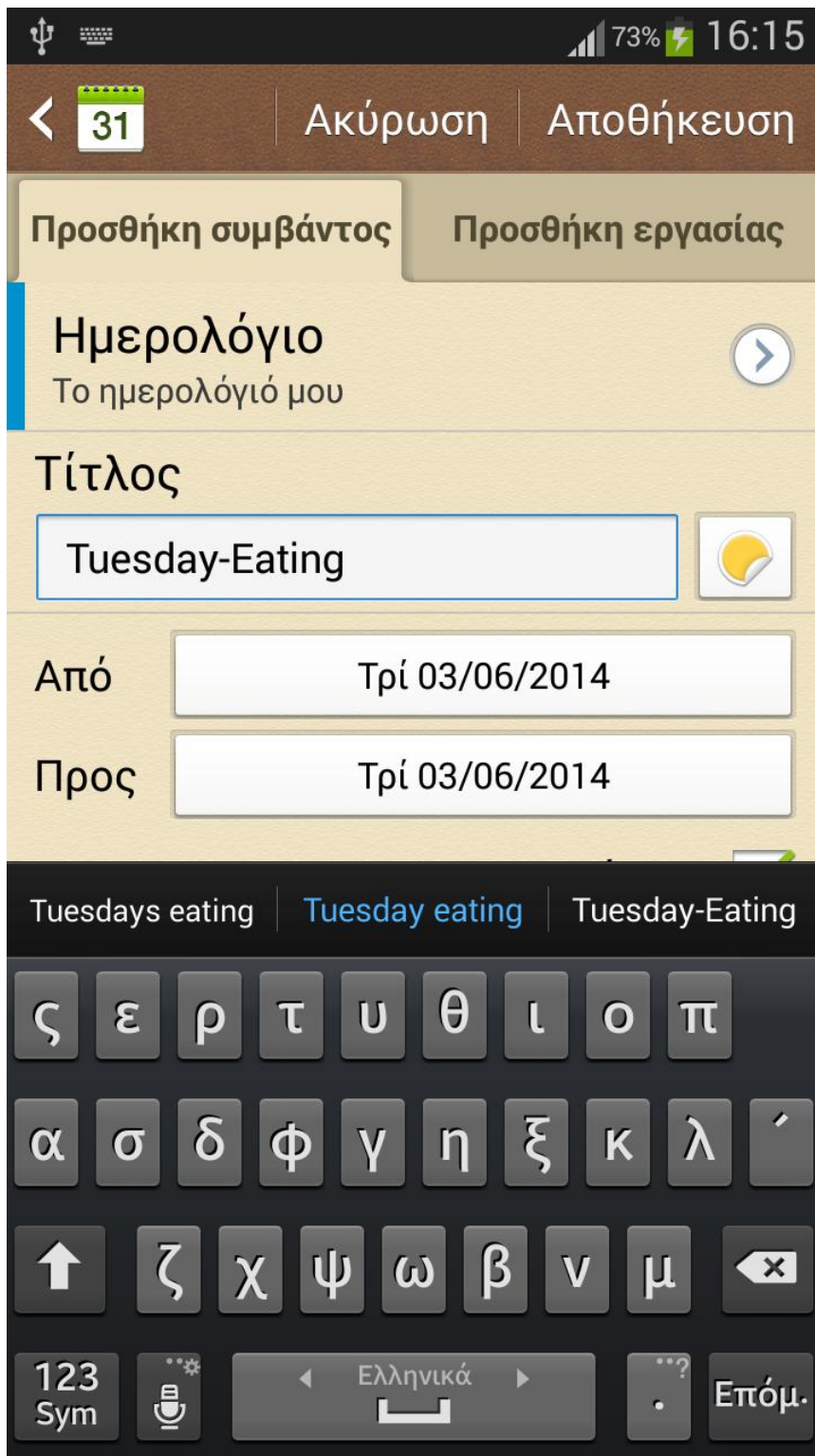
655

Long Press Day' s Name to add event to
Calendar

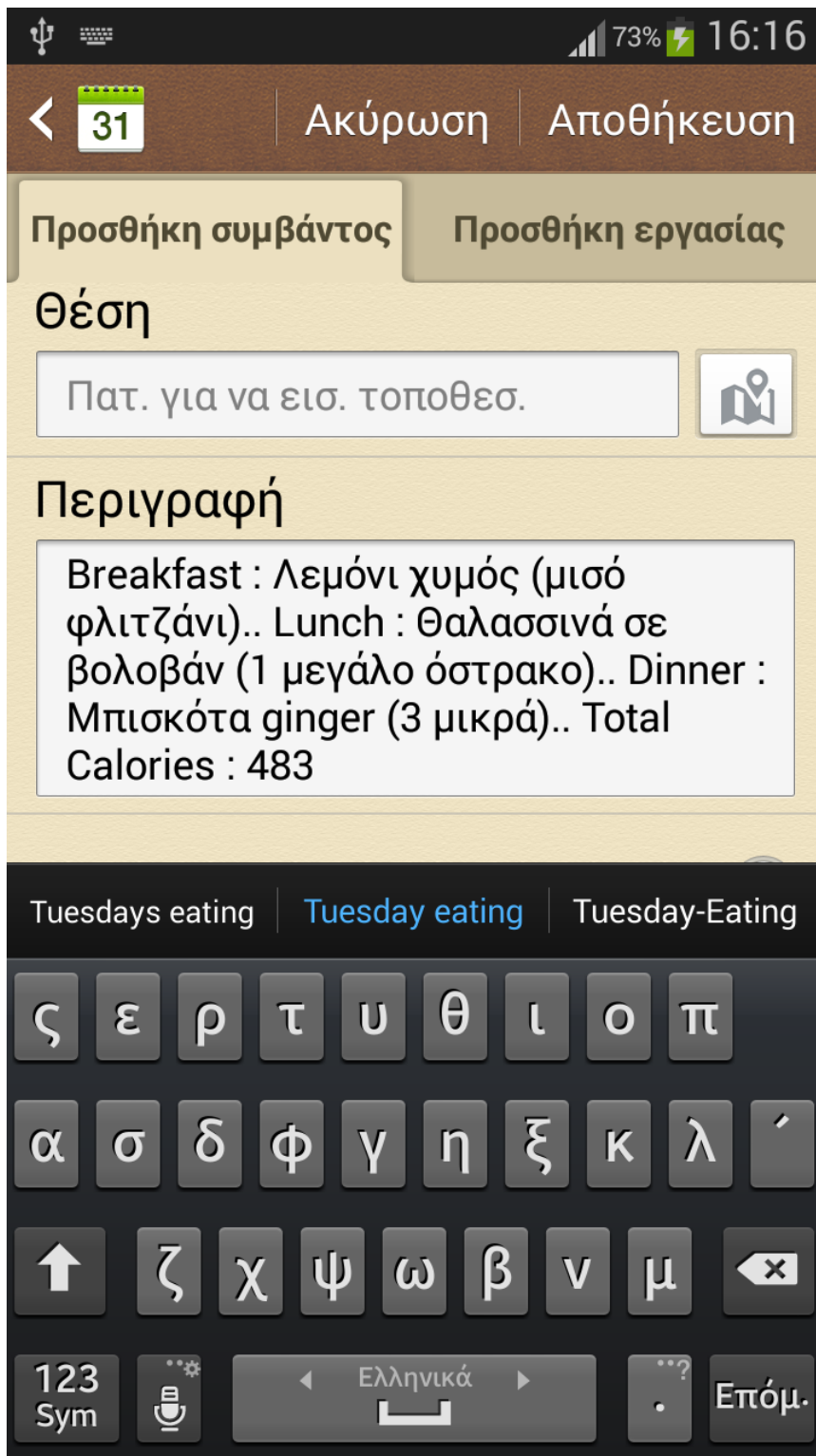
Εικόνα 39: Results Screen (1)



Εικόνα 40: Results Screen (2)



Εικόνα 41: Calendar Screen (1)

**Εικόνα 42: Calendar Screen (2)**

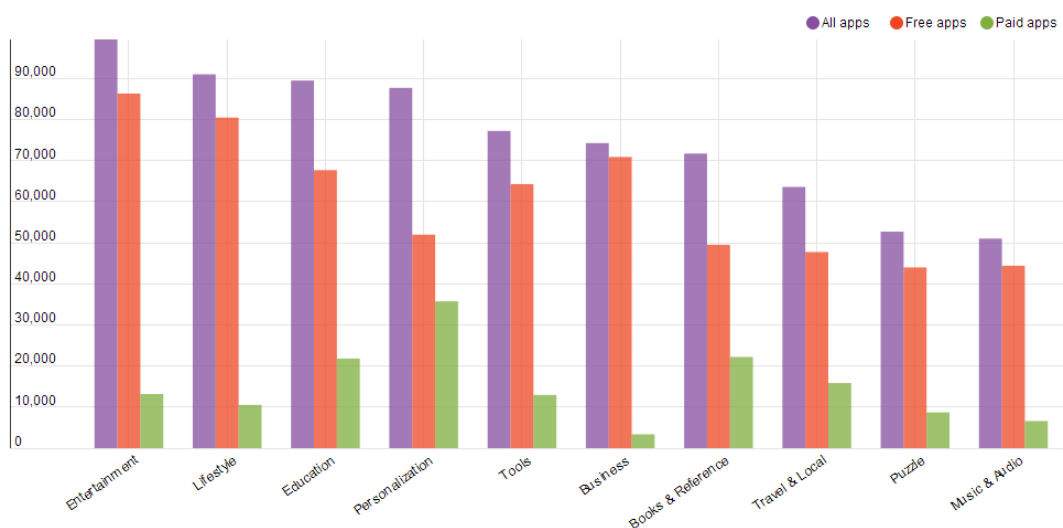
ΚΕΦΑΛΑΙΟ 6^ο

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ & ΠΕΡΙΛΗΨΗ

Το DietDroid αποτελεί μια ολοκληρωμένη εφαρμογή για τους χρήστες κινητών με *Android* λειτουργικό . Από την αρχή μέχρι το τέλος σχεδιάστηκε να είναι λειτουργικό , εύχρηστο , καθώς και λειτουργικό και απλό για τον χρήστη . Είχε ευχάριστο και προσεγμένο *user interface* και χρησιμοποιούσε πολλές τεχνικές και δυνατότητες του *Android*. Ο χρήστης εύκολα μπορεί να το εντάξει στη ζωή του σε καθημερινή βάση και να το χρησιμοποιεί . Χτίστηκε με πραγματικά στοιχεία , όσον αφορά και τους διατροφολογικούς τύπους και τα γεύματα και τις θερμίδες που αναφέρει .

Γενικά οι εφαρμογές που είναι ανεπτυγμένες σε *Android* αυξάνονται ραγδαία από την ημέρα κυκλοφορίας του λογισμικού αυτού . Ακριβώς επειδή το *Android* είναι *open source* , επειδή απευθύνεται σε όλους και επειδή δεν έχει μεγάλο κόστος για την συγγραφή και δημοσίευση εφαρμογών έχει γνωρίσει τόσο μεγάλη άνθηση. Οι προγραμματιστές δημιουργούν εφαρμογές για όλες τις δραστηριότητες που μπορεί να έχει ένας χρήστης , π.χ. για διασκέδαση , για υπηρεσίες , για την εργασία κ.α. Ο παρακάτω πίνακας δείχνει το συνολικό αριθμό πληρωμένων και δωρεάν εφαρμογών που έχουν δημιουργηθεί στο *Google Play* , χωρισμένες ανά κατηγορία χρήσης.

Top 10 Google Play categories



Εικόνα 43: Top 10 Google Play categories

Το σίγουρο είναι ότι μιλάμε για ένα σύστημα με μεγάλες δυνατότητες που συνεχώς εξελίσσεται με καινούργιες βιβλιοθήκες , άρα και δυνατότητες, και χρησιμοποιείται σε όλο και περισσότερες συσκευές , *smartphones* , *tablets* , *phablets* , *smart tvs* κ.α.

Τέλος , όπως αναφέρθηκε και παραπάνω σκοπός της εργασίας αυτής πέρα από το πρόγραμμα σαν πρόγραμμα είναι η εκμάθηση από μέρους μου και η εξοικείωση μου με το *Android* λογισμικό και τις βιβλιοθήκες που αυτό χρησιμοποιεί . Προσωπικός στόχος μου ήταν να χρησιμοποιήσω πολλά διαφορετικά στοιχεία του *Android* , να τα συνθέσω και να δημιουργήσω μια ολοκληρωμένη εφαρμογή για τους υποτιθέμενους πελάτες – χρήστες του *Android Market* . Πιστεύω ότι *DietDroid* αποτέλεσε μια πολύ καλή εφαρμογή που αποτελεί την αφετηρία στον κόσμο του *Android* .

6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η εφαρμογή παρότι είναι ολοκληρωμένη και δίνει ορθά αποτελέσματα στο χρήστη που τη χρησιμοποιεί θα μπορούσε στο μέλλον να έχει κάποιες επεκτάσεις .

- Θα μπορούσε να προστεθεί η επικοινωνία με ένα server ώστε να ανανεώνονται ανά τακτά χρονικά διαστήματα οι τροφές που είναι εγγεγραμμένες στη βάση .
- Θα μπορούσε να υπάρχει η επιλογή της επικοινωνίας με διαιτολόγο για να ελέγχει κατευθείαν τις ημερήσιες ή εβδομαδιαίες δίαιτες που επιλέγει ο χρήστης .
- Θα μπορούσαν να προστεθούν επιπλέον παράμετροι στη διαλογή των τροφίμων , π.χ. τροφές που δεν τρώει ο χρήστης , για να μην του εμφανίζει καθόλου αυτές τις τροφές . Αυτό βέβαια απαιτεί αλλαγή στη δομή της βάσης μας και ανάλυση όλων των τροφών σε συστατικά για να μπορεί να γίνει ο αποκλεισμός όπου δεν μας αρέσει ή δεν πρέπει να φάμε (π.χ. αλλεργία).
- Θα μπορούσαν να υπάρχει δυνατότητα καταχώρησης του τι τρώει ο χρήστης , να δηλώνει δηλαδή ο ίδιος την ώρα και τι έφαγε ανεξάρτητα της δίαιτας καθώς ίσως και το βάρος του , και όλα αυτά να μπορεί να τα παρακολουθεί ο διαιτολόγος απομακρυσμένα και να του κάνει συστάσεις .
- Θα μπορούν να ενταθούν σε ένα ευρετήριο τηλέφωνα και διευθύνσεις διαιτολόγων και κάποιων σχετικών με δίαιτα γραμμών υποστήριξης .
- Θα μπορούσαν να ενταθούν μέρη που μπορεί να καταναλώσει τις τροφές που του υποδεικνύονται ο χρήστης και είναι κοντά του . Αυτά τα μέρη μπορούν ίσως να βρεθούν άμα υπάρξει μια σύνδεση της εφαρμογής μας με κάποιο *social media* , π.χ. *FourSquare*.
- Θα μπορούσε να υπάρξει η δυνατότητα επικοινωνίας της εφαρμογής με τα *social media* π.χ. κοινοποίηση στο *facebook* της προτεινόμενης δίαιτας , με σκοπό την διαφήμιση και αναγνωσιμότητα της.

ΚΕΦΑΛΑΙΟ 7^ο

7.1 ΒΙΒΛΙΟΓΡΑΦΙΑ

7.1.1 ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Lauren Darcey , Shane Conder , «Μάθετε την Ανάπτυξη Εφαρμογών για το Android σε 24 Ωρες» , Εκδόσεις Μ.Γκιούρδας , Αθήνα 2012
2. Rogers Cadenhead , Laura Lemay , «Πλήρες Εγχειρίδιο της Java 2 (τρίτη έκδοση)» , Εκδόσεις Μ.Γκιούρδας , Αθήνα 2003

7.1.2 ΞΕΝΟΓΛΩΣΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

3. Wallace Jackson , «Android Apps for Absolute Beginners (second edition)» , Εκδόσεις Apress 2012
4. Jeff Friesen , «Learn Java for Android Development (second edition)» , Εκδόσεις Apress 2013
5. Satya Komatinemi , Dave MacLean0 , «Pro Android 4» , Εκδόσεις Apress 2012

7.1.3 ΗΛΕΚΤΡΟΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

6. <http://www.w3schools.com>
7. <http://droid-blog.net>
8. <http://www.flamingtext.com>
9. <http://examples.javacodegeeks.com>
10. <http://www.colorpicker.com>
11. <http://stackoverflow.com>
12. <https://thenewcircle.com>
13. <http://www.9android.net>
14. <http://www.androidhive.info>
15. <http://www.edumobile.org/android>
16. <http://androidexample.com>
17. <https://code.google.com/p/androidscreencast>
18. <http://www.shape.gr/diatrofi-adyntisma/sosti-diatrofi/prwino-gala-adyntisma-p2.html>
19. <http://www.tlife.gr/Article/FN/0-105-2149.html>
20. <http://www.atcare.gr/index.php/2008-12-01-09-36-40.html>
21. http://en.wikipedia.org/wiki/Android_version_history#Android_alpha
22. <http://www.digitallife.gr/android-history-timeline-32402>
23. <http://androidandme.com>
24. <http://android-developers.blogspot.gr>
25. <http://officialandroid.blogspot.gr>
26. <http://android-codes-examples.blogspot.gr>
27. <http://android-coding.blogspot.gr>
28. <https://code.google.com>

ΚΕΦΑΛΑΙΟ 8^ο

8.1 ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΕΣ

8.1.1 Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="theme_for_Activities">#FFFFFF</color>
    <!-- aspo xrwma -->
    <color name="menulayoutitem_textcolor">#523E27</color>
    <!-- brown -->
    <color name="menulayoutitem_shadowcolor">#FAA039</color>
    <!-- orange -->
    <color
name="menulayoutitem_background">@android:color/transparent</color>
    <!-- diafanis -->
    <color name="alertdialog_background">@android:color/transparent</color>
```

```

    <!-- diafanes -->
    <color name="helplayout_textcolor">#7f0000</color>
    <!-- red -->
    <color
name="helplayout_textview_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="inputtab_textview_textcolor">#523E27</color>
    <!-- brown -->
    <color name="inputtab_textview_shadowcolor">#FAA039</color>
    <!-- orange -->
    <color
name="inputtab_textview_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="inputtab_edittext_textcolor">#FA0000</color>
    <!-- red -->
    <color name="inputtab_calculatortab_edittext_border">#D6CED3</color>
    <!-- light grey -->
    <color name="inputtab_edittext_hint">#D6CED3</color>
    <!-- light grey -->
    <color
name="inputtab_imageswitcher_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color
name="inputtab_spinner_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="inputtab_spinner_textcolor">#FA0000</color>
    <!-- red -->
    <color name="inputtab_spinner_border">#D6CED3</color>
    <!-- light grey -->
    <color name="inputtab_spinner_hint">#D6CED3</color>
    <!-- light grey -->
    <color
name="inputtab_linearlayout_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color
name="inputtab_viewschwicher_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="tab_color_selected">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="tab_color_inactive">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="tab_color_pressed">#ffcc66</color>
    <!-- yellow -->
    <color name="tab_textcolor_focused">#523E27</color>
    <!-- brown -->
    <color name="tab_textcolor_default">#523E27</color>
    <!-- brown -->
    <color name="tab_textcolor_pressed">#ffcc66</color>
    <!-- yellow -->
    <color name="calculatortab_textview_textcolor">#523E27</color>
    <!-- brown -->
    <color name="calculatortab_shadowcolor">#FAA039</color>
    <!-- orange -->

```



```

    <color name="calculatortab_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="calculatortab_edittext_textcolor">#FA0000</color>
    <!-- red -->
    <color name="calculatortab_edittext_hint">#D6CED3</color>
    <!-- light grey -->
    <color
name="calculatortab_imagebutton">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="settings_textview_textcolor">#523E27</color>
    <!-- brown -->
    <color name="settings_textview_shadowcolor">#FAA039</color>
    <!-- orange -->
    <color
name="settings_textview_background">@android:color/transparent</color>
    <!-- diafanes -->
    <color name="settings_edittext_textcolor">#FA0000</color>
    <!-- red -->
    <color name="settings_edittext_hint">#FA0000</color>
    <!-- red -->
</resources>

```

8.1.2 Dimens.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <dimen name="menulayoutitem_textsize">50dp</dimen>
    <dimen name="helplayout_textsize">25dp</dimen>
    <dimen name="helplayout_textpadding_r">30dp</dimen>
    <dimen name="helplayout_textpadding_l">48dp</dimen>
    <dimen name="helplayout_textpadding_t_b">63dp</dimen>
    <dimen name="inputtab_textview_textsize">30dp</dimen>
    <dimen name="inputtab_edittext_textsize">25dp</dimen>
    <dimen name="inputtab_spinner_textsize">25dp</dimen>
    <dimen name="inputtab_viewswitcher_textsize">25dp</dimen>
    <dimen name="calculatortab_textview_textsize">30dp</dimen>
    <dimen name="calculatortab_edittext_textsize">25dp</dimen>
    <dimen name="calculatortab_imagebutton_margintop">20dp</dimen>
    <dimen name="settings_textview_textsize">30dp</dimen>
    <dimen name="settings_edittext_textsize">25dp</dimen>
</resources>

```

8.1.3 Strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="application_name">DietDroid</string> <!-- gia to label tou
icon -->
    <string name="help">Help Screen</string> <!-- gia to image description --
>
    <string name="menu">Main Menu Screen</string> <!-- gia to image
description -->
    <string name="find_diet">Find Diet Screen</string> <!-- gia to image
description -->
    <string name="settings">Settings Screen</string> <!-- gia to image
description -->
    <string name="loading">Loading Screen</string> <!-- gia to image
description -->
    <string name="help_item">Help</string>
    <string name="find_diet_item">Find Diet</string>
    <string name="settings_item">Settings</string>
    <string name="go_back_item">Go Back</string>
    <string name="exit_item">Exit</string>
    <string name="alertdialog_title">Confirm Exit</string> <!-- gia to alert
dialog -->
    <string name="alertdialog_message">Do you really want to exit?</string>
<!-- gia to alert dialog -->
    <string name="firsttab_title">Input Data</string> <!-- title input tab --
>
    <string name="secondtab_title">Calculators</string> <!-- title calc tab -
->
    <string name="baros">Input Weight (in kg)</string> <!-- input tab -->
    <string name="ypsos">Input Height (in cm)</string> <!-- input tab -->
    <string name="ilikia">Input Age (in years)</string> <!-- input tab -->
    <string name="fyllo">Input Sex</string> <!-- input tab -->
    <string name="drastiriothta_entos_douleias">Input physical activity at
work</string> <!-- input tab -->
    <string name="drastiriothta_ektos_douleias">Input physical activity out
of work</string> <!-- input tab -->
    <string-array name="eidh_drastiriothtwn">
        <item>light</item>
        <item>normal</item>
        <item>intensive</item>
    </string-array> <!-- input tab -->
    <string name="empty">null</string> <!-- input tab -->
    <string name="deiktis_mazas_swmatos">Body Mass Index (BMI)</string> <!--
calcu tab -->
    <string name="deiktis_mazas_swmatos_kategoria">BMI Category</string> <!--
calcu tab -->
    <string name="basikos_metabolismos">Basal Metabolic Rate (BMR -
kcal)</string> <!-- calcu tab -->
    <string name="fysiki_drastiriotita">Activity Factor</string> <!-- calcu
tab -->
    <string name="imerisies_thermidikes_anagkes">Daily Caloric Needs
(kcal)</string> <!-- calcu tab -->

```

```

    <string name="calcu_hint_1">weight/height are empty</string> <!-- calcu
tab -->
    <string name="calcu_hint_2">weight is empty</string> <!-- calcu tab -->
    <string name="calcu_hint_3">height is empty</string> <!-- calcu tab -->
    <string name="calcu_hint_4">BMI/sex/age are empty</string> <!-- calcu tab
-->
    <string name="calcu_hint_5">BMI/age are empty</string> <!-- calcu tab -->
    <string name="calcu_hint_6">sex/age are empty</string> <!-- calcu tab -->
    <string name="calcu_hint_7">BMI/sex are empty</string> <!-- calcu tab -->
    <string name="calcu_hint_8">age is empty</string> <!-- calcu tab -->
    <string name="calcu_hint_9">BMI is empty</string> <!-- calcu tab -->
    <string name="calcu_hint_10">sex is empty</string> <!-- calcu tab -->
    <string name="calcu_hint_11">activities/sex are empty</string> <!-- calcu
tab -->
    <string name="calcu_hint_12">activities are empty</string> <!-- calcu tab
-->
    <string name="calcu_hint_13">BMR/physical activity are empty</string> <!--
- calcu tab -->
    <string name="calcu_hint_14">physical activity is empty</string> <!--
calcu tab -->
    <string name="calcu_hint_15">BMR is empty</string> <!-- calcu tab -->
    <string name="elleipobaris">underweight</string> <!-- calcu tab -->
    <string name="fysiologikos">normal</string> <!-- calcu tab -->
    <string name="yperbaros">overweight</string> <!-- calcu tab -->
    <string name="first_bathmou_yperbaros">1o level obese</string> <!-- calcu
tab -->
    <string name="second_bathmou_yperbaros">2o level obese</string> <!--
calcu tab -->
    <string name="pathologika_yperbaros">pathologically obese</string> <!--
calcu tab -->
    <string name="megisto_baros">Max Weight</string> <!-- settings -->
    <string name="megisto_ypsos">Max Height</string> <!-- settings -->
    <string name="megisto_ilikia">Max Age</string> <!-- settings -->
    <string-array name="imeres_ebdomas">
        <item>Monday</item>
        <item>Tuesday</item>
        <item>Wednesday</item>
        <item>Thursday</item>
        <item>Friday</item>
        <item>Saturday</item>
        <item>Sunday</item>
    </string-array> <!-- result -->
</resources>

```

8.1.4 Styles.xml

```

<resources xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Application theme. -->
    <style name="AppTheme" parent="AppBaseTheme">
        <!-- All customizations that are NOT specific to a particular API-
level can go here. -->
    </style>

```

```

<style name="themeForAlertDialog">
    <item name="android:background">@color/alertdialog_background</item>
</style>

<style name="themeForActivities">
    <item name="android:background">@color/theme_for_Activities</item>
</style>

<style name="textviews_inputtab">
    <item
name="android:textColor">@color/inputtab_textview_textcolor</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">match_parent</item>
    <item name="android:orientation">vertical</item>
    <item name="android:shadowDx">2</item>
    <item name="android:shadowDy">2</item>
    <item
name="android:shadowColor">@color/inputtab_textview_shadowcolor</item>
    <item name="android:shadowRadius">4</item>
    <item
name="android:textSize">@dimen/inputtab_textview_textsize</item>
    <item
name="android:background">@color/inputtab_textview_background</item>
    <item name="android:gravity">center</item>
</style>

<style name="textviews_calculatortab">
    <item
name="android:textColor">@color/calculatortab_textview_textcolor</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">match_parent</item>
    <item name="android:orientation">vertical</item>
    <item name="android:shadowDx">2</item>
    <item name="android:shadowDy">2</item>
    <item
name="android:shadowColor">@color/calculatortab_shadowcolor</item>
    <item name="android:shadowRadius">4</item>
    <item
name="android:textSize">@dimen/calculatortab_textview_textsize</item>
    <item
name="android:background">@color/calculatortab_background</item>
    <item name="android:gravity">center</item>
</style>

<style name="textviews_settings">
    <item
name="android:textColor">@color/settings_textview_textcolor</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_width">match_parent</item>
    <item name="android:orientation">vertical</item>
    <item name="android:shadowDx">2</item>
    <item name="android:shadowDy">2</item>

```

```

        <item
name="android:shadowColor">@color/settings_textview_shadowcolor</item>
        <item name="android:shadowRadius">4</item>
        <item
name="android:textSize">@dimen/settings_textview_textsize</item>
        <item
name="android:background">@color/settings_textview_background</item>
        <item name="android:gravity">center</item>
    </style>

    <style name="edittexts_settings">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:inputType">number</item>
        <item name="android:maxLength">3</item>
        <item name="android:maxLines">1</item>
        <item
name="android:background">@layout/inputtab_calculatortab_edittext_borders</it
em>
        <item
name="android:textSize">@dimen/settings_edittext_textsize</item>
        <item
name="android:textColor">@color/settings_edittext_textcolor</item>
        <item name="android:gravity">center</item>
        <item
name="android:textColorHint">@color/settings_edittext_hint</item>
    </style>

    <style name="edittexts_inputtab">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:inputType">number</item>
        <item name="android:maxLength">3</item>
        <item name="android:maxLines">1</item>
        <item
name="android:background">@layout/inputtab_calculatortab_edittext_borders</it
em>
        <item
name="android:textSize">@dimen/inputtab_edittext_textsize</item>
        <item
name="android:textColor">@color/inputtab_edittext_textcolor</item>
        <item name="android:gravity">center</item>
        <item
name="android:textColorHint">@color/inputtab_edittext_hint</item>
    </style>

    <style name="edittexts_calculatortab">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:editable">>false</item>
        <item name="android:maxLines">1</item>

```

```

        <item
name="android:background">@layout/inputtab_calculatortab_edittext_borders</it
em>
        <item
name="android:textSize">@dimen/calculatortab_edittext_textsize</item>
        <item
name="android:textColor">@color/calculatortab_edittext_textcolor</item>
        <item name="android:gravity">center</item>
        <item
name="android:textColorHint">@color/calculatortab_edittext_hint</item>
        </style>

        <style name="text_type_1_results">
        <item
name="android:textColor">@color/calculatortab_textview_textcolor</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_width">match_parent</item>
        <item name="android:orientation">vertical</item>
        <item name="android:shadowDx">2</item>
        <item name="android:shadowDy">2</item>
        <item
name="android:shadowColor">@color/calculatortab_shadowcolor</item>
        <item name="android:shadowRadius">4</item>
        <item
name="android:textSize">@dimen/calculatortab_textview_textsize</item>
        <item
name="android:background">@color/calculatortab_background</item>
        <item name="android:gravity">center</item>
        </style>

        <style name="text_type_2_results">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:editable">>false</item>
        <item
name="android:background">@layout/inputtab_calculatortab_edittext_borders</it
em>
        <item
name="android:textSize">@dimen/calculatortab_edittext_textsize</item>
        <item
name="android:textColor">@color/calculatortab_edittext_textcolor</item>
        <item name="android:gravity">center</item>
        </style>
</resources>

```

8.1.4 Format_loadinglayout.xml

```

<html>
  <head>
    <script type='text/javascript'>
      function wh(){
        var img1=document.getElementsByTagName("img")[0];

```

```

        var img2=document.getElementsByTagName("img")[1];
        var att1=document.createAttribute("width");
        var att3=document.createAttribute("width");
        var att2=document.createAttribute("height");
        var att4=document.createAttribute("height");
        att1.value=window.screen.width;
        att3.value=window.screen.width;
        img1.setAttributeNode(att1);
        img1.setAttributeNode(att2);
        img2.setAttributeNode(att3);
        img2.setAttributeNode(att4);
    }
</script>
<style type='text/css'>
    html,body {margin: 0;padding: 0;width: 100%;height:
100%;}
    html {display: table;}
    body {display: table-cell;vertical-align:
middle;horizontal-align: middle;text-align: center;}
</style>
</head>
<body onload='wh()'>
<img align='middle' src='file:///android_asset/loadingdroid.gif' >
</img>
<img align='middle' src='file:///android_asset/loading.gif' >
</img>
</body>
</html>

```

8.1.5 ExitDroid.java

```

package com.example.dietdroid;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Handler;
import android.widget.Toast;

public class ExitDroid {

    public ExitDroid(Activity act) {
        final Activity activity = act;
        final Context con = act.getApplicationContext();
        AlertDialog.Builder alertDialogBuilder_1 = new
AlertDialog.Builder(act);
        act.setTheme(R.style.themeForAlertDialog);
        alertDialogBuilder_1.setTitle(R.string.alertdialog_title);
        alertDialogBuilder_1.setMessage(R.string.alertdialog_message);
        alertDialogBuilder_1.setIcon(R.drawable.exit_menu_icon); //

```

72*72

```

        // Setting Positive "Yes" Btn
        alertDialogBuilder_1.setPositiveButton("YES",
            new DialogInterface.OnClickListener() {
                public void onClick(final DialogInterface
dialog, int which) {
                    // Write your code here to execute
                    after dialog
                    Handler handler_2 = new Handler();
                    Runnable runnable_2 = new Runnable() {
                        public void run() {
                            activity.finish();
                            System.exit(0);
                        }
                    };
                    handler_2.postDelayed(runnable_2,
500);
                    Toast.makeText(con, "You clicked on
YES",
                        Toast.LENGTH_SHORT).show();
                }
            });
        // Setting Negative "NO" Btn
        alertDialogBuilder_1.setNegativeButton("NO",
            new DialogInterface.OnClickListener() {
                public void onClick(final DialogInterface
dialog, int which) {
                    // Write your code here to execute
                    after dialog
                    Handler handler_3 = new Handler();
                    Runnable runnable_3 = new Runnable() {
                        public void run() {
                            dialog.cancel();
                        }
                    };
                    handler_3.postDelayed(runnable_3,
500);
                    Toast.makeText(con, "You clicked on
NO",
                        Toast.LENGTH_SHORT).show();
                }
            });
        AlertDialog alertDialog_1 = alertDialogBuilder_1.create();
        // Showing Alert Dialog
        alertDialog_1.show();
    }
}

```


8.1.6 FullScreen.java

```
package com.example.dietdroid;

import android.app.Activity;
import android.view.Window;
import android.view.WindowManager;

public class FullScreen{

    public FullScreen(Activity act){

        act.requestWindowFeature(Window.FEATURE_NO_TITLE);

        act.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                                WindowManager.LayoutParams.FLAG_FULLSCREEN);

    }

}
```

8.1.7 InputStreamToString.java

```
package com.example.dietdroid;

import java.io.DataInputStream;
import java.io.IOException;
import java.io.InputStream;

public class InputStreamToString {
    StringBuffer sBuffer = null;

    @SuppressWarnings("deprecation")
    public InputStreamToString(InputStream instr) throws IOException{
        sBuffer = new StringBuffer();
        DataInputStream dataIO = new DataInputStream(instr);
        String strLine = null;
        while((strLine = dataIO.readLine()) !=null){
            sBuffer.append(strLine + "\n");
        }
        dataIO.close();
        instr.close();
    }
}
```

8.1.8 JumpToActivity.java

```
package com.example.dietdroid;
```

```
import android.app.Activity;
import android.content.Intent;
import android.os.Handler;

public class JumpToActivity {
    Activity oldact;
    Class<?> newact;

    public JumpToActivity(Activity oldActivity, Class<?> newActivity, long
delay){

        oldact = oldActivity;
        newact = newActivity;
        Handler handler_1 = new Handler();
        Runnable runnable_1 = new Runnable()
        {
            public void run(){
                oldact.startActivity(new
Intent(oldact,newact));
                oldact.finish();
            }
        };
        handler_1.postDelayed(runnable_1, 13000);
    }
}
```

8.1.9 Prwina.java

```
package com.example.dietdroid;

public class Prwina {
    // private variables
    int _id_prwina;
    String _name_prwina;
    int _thermides_prwina;
```

```
// Empty constructor
public Prwina() {
}

// constructor
public Prwina(int id_prwina, String name_prwina, int thermides_prwina)
{
    this._id_prwina = id_prwina;
    this._name_prwina = name_prwina;
    this._thermides_prwina = thermides_prwina;
}

// constructor
public Prwina(String name_prwina, int thermides_prwina) {
    this._name_prwina = name_prwina;
    this._thermides_prwina = thermides_prwina;
}

// getting ID
public int getIDPrwina() {
    return this._id_prwina;
}

// setting id
public void setIDPrwina(int id) {
    this._id_prwina = id;
}

// getting name
public String getNamePrwina() {
    return this._name_prwina;
}

// setting name
public void setNamePrwina(String name_prwina) {
    this._name_prwina = name_prwina;
}

// getting thermides
public int getThermidesPrwina() {
    return this._thermides_prwina;
}

// setting thermides
public void setThermidesPrwina(int thermides_prwina) {
    this._thermides_prwina = thermides_prwina;
}
```

8.1.10 Mesimeriana.java

```
package com.example.dietdroid;
Ανάπτυξη Android Εφαρμογής «Δίαιτα (DietDroid)»
```

```
public class Mesimeriana {
    // private variables
    int _id_mesimeriana;
    String _name_mesimeriana;
    int _thermides_mesimeriana;

    // Empty constructor
    public Mesimeriana() {

    }

    // constructor
    public Mesimeriana(int id_mesimeriana, String name_mesimeriana, int
thermides_mesimeriana) {
        this._id_mesimeriana = id_mesimeriana;
        this._name_mesimeriana = name_mesimeriana;
        this._thermides_mesimeriana = thermides_mesimeriana;
    }

    // constructor
    public Mesimeriana(String name_mesimeriana, int thermides_mesimeriana)
{
        this._name_mesimeriana = name_mesimeriana;
        this._thermides_mesimeriana = thermides_mesimeriana;
    }

    // getting ID
    public int getIDMesimeriana() {
        return this._id_mesimeriana;
    }

    // setting id
    public void setIDMesimeriana(int id) {
        this._id_mesimeriana = id;
    }

    // getting name
    public String getNameMesimeriana() {
        return this._name_mesimeriana;
    }

    // setting name
    public void setNameMesimeriana(String name_mesimeriana) {
        this._name_mesimeriana = name_mesimeriana;
    }

    // getting thermides
    public int getThermidesMesimeriana() {
        return this._thermides_mesimeriana;
    }

    // setting thermides
```

```
    public void setThermidesMesimeriana(int thermides_mesimeriana) {
        this._thermides_mesimeriana = thermides_mesimeriana;
    }
}
```

8.1.11 Bradyna.java

```
package com.example.dietdroid;

public class Bradyna {
    // private variables
    int _id_bradyana;
    String _name_bradyana;
    int _thermides_bradyana;

    // Empty constructor
    public Bradyna() {

    }

    // constructor
    public Bradyna(int id_bradyana, String name_bradyana, int
thermides_bradyana) {
        this._id_bradyana = id_bradyana;
        this._name_bradyana = name_bradyana;
        this._thermides_bradyana = thermides_bradyana;
    }

    // constructor
    public Bradyna(String name_bradyana, int thermides_bradyana) {
        this._name_bradyana = name_bradyana;
        this._thermides_bradyana = thermides_bradyana;
    }

    // getting ID
    public int getIDBradyna() {
        return this._id_bradyana;
    }

    // setting id
    public void setIDBradyna(int id) {
        this._id_bradyana = id;
    }

    // getting name
    public String getNameBradyna() {
        return this._name_bradyana;
    }

    // setting name
    public void setNameBradyna(String name_bradyana) {
        this._name_bradyana = name_bradyana;
    }
}
```

```

    }

    // getting thermides
    public int getThermidesBradyna() {
        return this._thermides_bradyna;
    }

    // setting thermides
    public void setThermidesBradyna(int thermides_bradyna) {
        this._thermides_bradyna = thermides_bradyna;
    }
}

```

8.1.12 DatabaseHandler.java

```

package com.example.dietdroid;

import java.util.ArrayList;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DatabaseHandler extends SQLiteOpenHelper {

    // All Static variables
    // Database Version
    private static final int DATABASE_VERSION = 1;

    // Database Name
    private static final String DATABASE_NAME = "DietDroidDatabase";

    // table names
    private static final String TABLE_PRWINA = "Prwina";
    private static final String TABLE_MESIMERIANA = "Mesimeriana";
    private static final String TABLE_BRADYNA = "Bradyna";

    // Table Columns names
    private static final String KEY_ID_PRWINA = "id_prwina";
    private static final String KEY_GEUMA_PRWINA = "geuma_prwina";
    private static final String KEY_THERMIDES_PRWINA = "thermides_prwina";

    private static final String KEY_ID_MESIMERIANA = "id_mesimeriana";
    private static final String KEY_GEUMA_MESIMERIANA =
"geuma_mesimeriana";
    private static final String KEY_THERMIDES_MESIMERIANA =
"thermides_mesimeriana";

```

```

private static final String KEY_ID_BRADYNA = "id_bradydna";
private static final String KEY_GEUMA_BRADYNA = "geuma_bradydna";
private static final String KEY_THERMIDES_BRADYNA =
"thermidides_bradydna";

public DatabaseHandler(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

// Creating Tables
@Override
public void onCreate(SQLiteDatabase db) {

    String CREATE_PRWINA_TABLE = "CREATE TABLE IF NOT EXISTS "
        + TABLE_PRWINA + "(" + KEY_ID_PRWINA + " INTEGER
PRIMARY KEY,"
        + KEY_GEUMA_PRWINA + " TEXT," +
KEY_THERMIDES_PRWINA
        + " INTEGER " + ")";

    String CREATE_MESIMERIANA_TABLE = "CREATE TABLE IF NOT EXISTS "
        + TABLE_MESIMERIANA + "(" + KEY_ID_MESIMERIANA
        + " INTEGER PRIMARY KEY," + KEY_GEUMA_MESIMERIANA +
" TEXT,"
        + KEY_THERMIDES_MESIMERIANA + " INTEGER " + ")";

    String CREATE_BRADYNA_TABLE = "CREATE TABLE IF NOT EXISTS "
        + TABLE_BRADYNA + "(" + KEY_ID_BRADYNA
        + " INTEGER PRIMARY KEY," + KEY_GEUMA_BRADYNA + "
TEXT,"
        + KEY_THERMIDES_BRADYNA + " INTEGER " + ")";

    db.execSQL(CREATE_PRWINA_TABLE);
    db.execSQL(CREATE_MESIMERIANA_TABLE);
    db.execSQL(CREATE_BRADYNA_TABLE);
}

// Upgrading database
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    // Drop older tables if existed
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_PRWINA);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_MESIMERIANA);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_BRADYNA);

    // Create tables again
    onCreate(db);
}

// Adding new PRWINA
void addPrwina(Prwina prwina) {
    SQLiteDatabase db = this.getWritableDatabase();

```

```

        ContentValues values_prwina = new ContentValues();
        values_prwina.put(KEY_GEUMA_PRWINA, prwina.getNamePrwina());
        values_prwina.put(KEY_THERMIDES_PRWINA,
prwina.getThermidesPrwina());

        // Inserting Row
        db.insert(TABLE_PRWINA, null, values_prwina);
        db.close(); // Closing database connection
    }

    // Adding new MESIMERIANA
    void addMesimeriana(Mesimeriana mesimeriana) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values_mesimeriana = new ContentValues();
        values_mesimeriana.put(KEY_GEUMA_MESIMERIANA,
            mesimeriana.getNameMesimeriana());
        values_mesimeriana.put(KEY_THERMIDES_MESIMERIANA,
            mesimeriana.getThermidesMesimeriana());

        // Inserting Row
        db.insert(TABLE_MESIMERIANA, null, values_mesimeriana);
        db.close(); // Closing database connection
    }

    // Adding new MESIMERIANA
    void addBradyna(Bradyna bradyna) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values_bradyna = new ContentValues();
        values_bradyna.put(KEY_GEUMA_BRADYNA, bradyna.getNameBradyna());
        values_bradyna
            .put(KEY_THERMIDES_BRADYNA,
bradyna.getThermidesBradyna());

        // Inserting Row
        db.insert(TABLE_BRADYNA, null, values_bradyna);
        db.close(); // Closing database connection
    }

    // Getting PRWINA
    public List<Prwina> getPrwina() {
        List<Prwina> prwinaList = new ArrayList<Prwina>();

        // Select Query
        String selectQuery = "SELECT * FROM " + TABLE_PRWINA + " WHERE
"
            + KEY_THERMIDES_PRWINA + " <= (0.3* 0.9* "
            + FindDietActivity.draft_5 + " )";

        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);

```



```

        int p = 0;
        // looping through all rows and adding to list
        if (cursor.moveToFirst()) {
            do {p++;
                Prwina prwina = new Prwina();

                prwina.setIDPrwina(Integer.parseInt(cursor.getString(0)));
                Log.i("prwina", cursor.getString(0));
                prwina.setNamePrwina(cursor.getString(1));
                Log.i("prwina", cursor.getString(1));

                prwina.setThermidesPrwina(Integer.parseInt(cursor.getString(2)));
                Log.i("prwina", cursor.getString(2));
                // Adding prwina to list
                prwinaList.add(prwina);

            } while (cursor.moveToNext());
            Log.i("posa_prwina", p+"");
        }

        // return prwina list
        return prwinaList;
    }

    // Getting MESIMERIANA
    public List<Mesimeriana> getMesimeriana() {
        List<Mesimeriana> mesimerianaList = new
ArrayList<Mesimeriana>();
        // Select Query
        String selectQuery = "SELECT * FROM " + TABLE_MESIMERIANA + "
WHERE "
                + KEY_THERMIDES_MESIMERIANA + " <= (0.5* 0.9* "
                + FindDietActivity.draft_5 + " )";

        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);
        int m = 0;
        // looping through all rows and adding to list
        if (cursor.moveToFirst()) {
            do {m++;
                Mesimeriana mesimeriana = new Mesimeriana();

                mesimeriana.setIDMesimeriana(Integer.parseInt(cursor
                    .getString(0)));
                Log.i("mesimeriana", cursor.getString(0));

                mesimeriana.setNameMesimeriana(cursor.getString(1));
                Log.i("mesimeriana", cursor.getString(1));

                mesimeriana.setThermidesMesimeriana(Integer.parseInt(cursor
                    .getString(2)));
                Log.i("mesimeriana", cursor.getString(2));
                // Adding mesimeriana to list

```

```

        mesimerianaList.add(mesimeriana);
    } while (cursor.moveToNext());
    Log.i("posa_mesimeriana", m+"");
}

// return mesimeriana list
return mesimerianaList;
}

// Getting BRADYNA
public List<Bradyna> getBradyna() {
    List<Bradyna> bradynaList = new ArrayList<Bradyna>();
    // Select Query
    String selectQuery = "SELECT * FROM " + TABLE_BRADYNA + " WHERE
    "
        + KEY_THERMIDES_BRADYNA + " <= (0.2* 0.9* "
        + FindDietActivity.draft_5 + " )";

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    int b = 0;
    // looping through all rows and adding to list
    if (cursor.moveToFirst()) {
        do {b++;
            Bradyna bradyna = new Bradyna();

            bradyna.setIDBradyna(Integer.parseInt(cursor.getString(0)));
            Log.i("bradyna", cursor.getString(0));
            bradyna.setNameBradyna(cursor.getString(1));
            Log.i("bradyna", cursor.getString(1));
            bradyna.setThermidesBradyna(Integer.parseInt(cursor
                .getString(2)));
            Log.i("bradyna", cursor.getString(2));
            // Adding bradyna to list
            bradynaList.add(bradyna);
        } while (cursor.moveToNext());
        Log.i("posa_bradyna", b+"");
    }

    // return mesimeriana list
    return bradynaList;
}
}
}

```

8.1.13 MenuActivity.java (τμήμα του)

```

db.addPrwina(new Prwina("Τοστ με αβγό", 255));
db.addPrwina(new Prwina("Τοστ ζαμπόντυρί", 400));
db.addPrwina(new Prwina("Τοστ με 1 φέτα ζαμπόν", 270));
db.addPrwina(new Prwina(
    "Ρυζόγαλο + Καφές αμερικανικός ή γαλλικός σκέτος (1
    φλ.)", 302));

```

Ανάπτυξη Android Εφαρμογής «Δίαιτα (DietDroid)»

```

db.addPrwina(new Prwina("Τοστ με μπιφτέκι", 440));
db.addPrwina(new Prwina("Τοστ με 1 λεπτή φέτα χοιρομέρι", 435));
db.addPrwina(new Prwina("Αυγά Ομελέτα και τυρί", 470));
db.addPrwina(new Prwina("Αχλαδόμηλο μαρμελάδα (100γρ)", 300));
db.addPrwina(new Prwina(
    "Γκούντα τυρί (100γρ) + γαλοπούλα βραστή στήθος
(275γρ)", 625));
db.addPrwina(new Prwina(
    "Δημητριακά με μέλι και ξηρούς καρπούς (100γρ.) ",
390));
db.addPrwina(new Prwina(
    "Δημητριακά με σταφίδες και χουρμάδες (100γρ.)",
455));
db.addPrwina(new Prwina("Δημητριακά Musli (100γρ.)", 390));
db.addPrwina(new Prwina("Δημητριακά με πίτουρο (1 φλ.)", 210));
db.addPrwina(new Prwina(
    "Ένταμ τυρί (100 γρ.) + Ψωμάκι σκληρό για σάντουιτς
50γρ.", 496));
db.addPrwina(new Prwina(
    "Γάλα συμπυκνωμένο ζαχαρούχο (1/2 φλιτζ. τσαγιού)",
540));
db.addPrwina(new Prwina("Γαλατόπιτα", 300));
db.addPrwina(new Prwina("Γάλα με φρούτα (μιλκσέικ 1 ποτήρι)",
190));
db.addPrwina(new Prwina(
    "Γάλα κατσικίσιο (1 φλιτζ. τσαγιού) + Φρυγανιές
απλές μεγάλες (3) ",
454));
db.addPrwina(new Prwina("Γάλα αγελάδος εβαπορέ (1/2 φλ.
τσαγιού)", 190));
db.addPrwina(new Prwina(
    "Έμενταλ τυρί (100 γρ.) + Ζαμπόν ημίπαχο (100 γρ.)
+ Ψωμάκι με σταφίδες 50γρ. + Λεμόνι χυμός (μισό φλιτζάνι)",
883));
db.addPrwina(new Prwina(
    "Γραβιέρα τυρί (100 γρ.) + Ψωμάκι σκληρό για
σάντουιτς 50γρ.",
596));
db.addPrwina(new Prwina(
    "Καφές με ένα κ. γλυκού ζάχαρη + Φρυγανιές απλές
μεγάλες (2) ",
198));
db.addPrwina(new Prwina("Κέικ με σοκολάτα (1 κομμάτι 50γρ.)",
325));
db.addPrwina(new Prwina("Κέικ φρούτων (1 κομμάτι 50γρ.)", 300));

```

```

        db.addPrwina(new Prwina(
            "Κεφαλοτύρι (100 γρ.) + Ψωμάκι σκληρό για σάντουιτς
50γρ.", 556));
        db.addPrwina(new Prwina("Κακάο ρόφημα (2+2 κουτ.
κακάο/ζάχαρη) ", 205));
        db.addPrwina(new Prwina("Κουάκερ κρέμα (1 μπολ)", 209));
        db.addPrwina(new Prwina("Κουφέτα με αμύγδαλα (100γρ.) ", 458));
        db.addPrwina(new Prwina("Κρουασάν με σοκολάτα (1 μεγάλο
75γρ.) ", 300));
        db.addPrwina(new Prwina("Κωκ (1 μεγάλο)", 360));
        db.addPrwina(new Prwina("Λεμονάδα (2 λεμόνια με 2 κ. γλ.
ζάχαρη)", 100));
        db.addPrwina(new Prwina("Λεμόνι χυμός (μισό φλιτζάνι)", 33));
        db.addPrwina(new Prwina(
            "Λουκουμάδες Καφές αμερικανικός ή γαλλικός σκέτος
(1 φλ.)", 312));
        db.addPrwina(new Prwina("Μαρμελάδα (1 κουτ. γλυκού)", 55));
        db.addPrwina(new Prwina("Μπέικον καπνιστό (100 γρ.)", 770));
        db.addPrwina(new Prwina("Μπέικον τηγανητό (3 φέτες) ", 360));
        db.addPrwina(new Prwina(
            "Μπέικον αλατισμένο (100 γρ.) + Ψωμάκι με σταφίδες
50γρ.", 950));
        db.addPrwina(new Prwina("Μήλα στρούντελ με αμύγδαλα/καρύδια",
460));
        db.addPrwina(new Prwina("Μηλόπιτα", 395));
        db.addPrwina(new Prwina("Μπουγάτσα", 360));
        db.addPrwina(new Prwina("Ντόνατς (1)", 180));
        db.addPrwina(new Prwina("Πιτάκια με μαρμελάδα (1 μικρό)", 200));
        db.addPrwina(new Prwina("Τσουρέκι μπριός (1 ατομικό) ", 230));
        db.addPrwina(new Prwina("Τυρί κότατζ (170 γρ.)", 190));
        db.addPrwina(new Prwina("Τυρόπιτα", 37));
        db.addPrwina(new Prwina("Φιστικοβούτυρο (100 γρ.)", 660));
        db.addPrwina(new Prwina("Χάμπουργκερ (1 μεγάλο)", 440));
        db.addPrwina(new Prwina("Χάμπουργκερ (1 μικρό)", 275));
        db.addPrwina(new Prwina("Χαλβάς εμπορίου με βανίλια (100 γρ.)",
506));
        db.addPrwina(new Prwina("Χορτόπιτα (1 μέτριο κομμάτι)", 375));

```

Κώδικας εισαγωγής πρωινών στη βάση δεδομένων.

```

db.addMesimeriana(new Mesimeriana("Αμπελοφάσουλα σαλάτα", 340));
        db.addMesimeriana(new Mesimeriana("Αρνί με αρακά", 800));
        db.addMesimeriana(new Mesimeriana("Αρνί με κολοκυθάκια", 740));
        db.addMesimeriana(new Mesimeriana("Αρνί μπριζόλες πανέ", 700));

```

```

db.addMesimeriana(new Mesimeriana("Αρνί παϊδάκια σχάρας", 370));
db.addMesimeriana(new Mesimeriana("Αρνί παϊδάκια τηγανητά",
610));
db.addMesimeriana(new Mesimeriana(
    "Αρνί φρικασέ + Ανθότυρο (100 γρ.) + Ελιές μαύρες
(200 γρ.) + Ψωμί άσπρο (2 λεπτές φέτες)",
1530));
db.addMesimeriana(new Mesimeriana("Ξιφίας ψητός", 285));
db.addMesimeriana(new Mesimeriana(
    "Γιουβέτσι με αρνί + Γκαζόζα (1 ποτήρι)", 985));
db.addMesimeriana(new Mesimeriana(
    "Γκούλας με κρέας + Γκούντα τυρί (μερίδα)", 1110));
db.addMesimeriana(new Mesimeriana("Χυλοπίτες με τυρί και
ντομάτα", 570));
db.addMesimeriana(new Mesimeriana("Χταπόδι με σπανάκι", 460));
db.addMesimeriana(new Mesimeriana("Χοιρινό με σέλινο", 750));
db.addMesimeriana(new Mesimeriana("Χοιρινό σουβλας στήθος",
435));
db.addMesimeriana(new Mesimeriana("Χοιρινό μπριζόλα σχάρας
άπαχη", 350));
db.addMesimeriana(new Mesimeriana("Χάμπουργκερ (1 μεγάλο)",
440));
db.addMesimeriana(new Mesimeriana("Χυλοπίτες με κοτόπουλο",
580));
db.addMesimeriana(new Mesimeriana(
    "Θαλασσινά σε Κοκίγ με κρέμα γάλακτος και τυρί (1
μεγάλο όστρακο)",
310));
db.addMesimeriana(new Mesimeriana("Ζυμαρικά απλά με κιμά",
740));
db.addMesimeriana(new Mesimeriana("Ζυμαρικά απλά με σάλτσα
μανιτάρια",
680));
db.addMesimeriana(new Mesimeriana("Ζυμαρικά με βούτυρο & τυρί",
535));
db.addMesimeriana(new Mesimeriana("Βοδινό φιλέτο σχάρας", 300));
db.addMesimeriana(new Mesimeriana("Βοδινό ψητό φούρνου", 440));
db.addMesimeriana(new Mesimeriana("Γαρίδες γιουβέτσι", 540));
db.addMesimeriana(new Mesimeriana("Γαρίδες τηγανητές", 330));
db.addMesimeriana(new Mesimeriana("Γαλοπούλα γεμιστή", 495));
db.addMesimeriana(new Mesimeriana(
    "Γκούλας με κρέας + Ψωμί άσπρο (2 λεπτές φέτες)",
1090));
db.addMesimeriana(new Mesimeriana(
    "Εσκαλόπ με ζαμπόν & πατάτες τηγανητές", 650));

```

```

db.addMesimeriana(new Mesimeriana(
    "Θαλασσινά σε βολοβάν (1 μεγάλο όστρακο)", 395));
db.addMesimeriana(new Mesimeriana(
    "Κόκορας κρασάτος (μερίδα) + Κόκα Κόλα (1 ποτήρι)",
490));
db.addMesimeriana(new Mesimeriana("Κεφτέδες (4 κανονικοί)",
460));
db.addMesimeriana(new Mesimeriana(
    "Κανελόνια με κιμά γκρατινέ (4 μεγάλα)", 1700));
db.addMesimeriana(new Mesimeriana(
    "Καρμπονάρα σπαγγέτι (μερίδα 250γρ.)", 840));
db.addMesimeriana(new Mesimeriana("Κότα ψητή σουβλας (1
μερίδα)", 250));
db.addMesimeriana(new Mesimeriana(
    "Κοτόσουπα με ρύζι αυγολέμονο (1 μερίδα 350γρ.)",
290));
db.addMesimeriana(new Mesimeriana("Κουκιά φρέσκα λαδερά", 430));
db.addMesimeriana(new Mesimeriana("Κουνέλι κοκκινιστό (μερίδα
260γρ.)",
    800));
db.addMesimeriana(new Mesimeriana(
    "Κριθαράκι με σάλτσα ντομάτας (μερίδα 220γρ.)",
520));
db.addMesimeriana(new Mesimeriana("Κουνουπίδι γιαχνί (μερίδα
260γρ.)",
    190));
db.addMesimeriana(new Mesimeriana(
    "Λουκάνικα μοσχαρίσια ψητά (8 πιρουιές)", 528));
db.addMesimeriana(new Mesimeriana("Λυθρίνι ψητό χωρίς λάδι (1
κιλό)",
    1000));
db.addMesimeriana(new Mesimeriana("Μοσχάρι κοκκινιστό με
πατάτες", 710));
db.addMesimeriana(new Mesimeriana("Μοσχάρι σνίτσελ με πουρέ",
600));
db.addMesimeriana(new Mesimeriana("Μοσχάρι φιλέτο σχάρας",
250));
db.addMesimeriana(new Mesimeriana("Μοσχάρι φιλέτο με πατάτες",
450));
db.addMesimeriana(new Mesimeriana("Μοσχάρι βραστό", 255));
db.addMesimeriana(new Mesimeriana(
    "Μπακαλιάρος τηγανητός με 1 κουταλιά σούπας
σκορδαλιά", 520));
db.addMesimeriana(new Mesimeriana(
    "Μπιφτέκια ατμού χωρίς ψωμί (1 μεγάλο )", 320));

```

```

        db.addMesimeriana(new Mesimeriana("Ντομάτες γεμιστές με κιμά
(1)", 315));
        db.addMesimeriana(new Mesimeriana("Ντολμάδες με λάχανο
αβγολέμονο", 400));

```

Κώδικας εισαγωγής μεσημεριανών στη βάση δεδομένων.

```

        db.addBradyna(new Bradyna("Αχλάδι κομπόστα (1 μέτριο)", 165));
        db.addBradyna(new Bradyna("Αχλάδι τάρτα", 385));
        db.addBradyna(new Bradyna("Αυγά με ντομάτες (3 κ. σούπας)",
165));
        db.addBradyna(new Bradyna("Βερίκοκα κομπόστα (2 κομμάτια
σιρόπι)", 135));
        db.addBradyna(new Bradyna("Βερίκοκα τάρτα", 360));
        db.addBradyna(new Bradyna("Βατόμουρα φρέσκα (ωμά 1 κιλό)",
600));
        db.addBradyna(new Bradyna("Βατόμουρα τάρτα", 350));
        db.addBradyna(new Bradyna("Γαλατόπιτα", 300));
        db.addBradyna(new Bradyna("Γκρέιπφρουτ (1 κομμάτι)", 55));
        db.addBradyna(new Bradyna("Γιαρμάδες (1 κιλό)", 440));
        db.addBradyna(new Bradyna("Γραβιέρα τυρί (μερίδα)", 240));
        db.addBradyna(new Bradyna("Δαμάσκηνα κομπόστα (5 κομμάτια)",
190));
        db.addBradyna(new Bradyna("Δαμάσκηνα ωμά (1 κιλό)", 500));
        db.addBradyna(new Bradyna("Έλιές πράσινες μέτριες (100 γρ.)",
105));
        db.addBradyna(new Bradyna("Εκλέρ σοκολάτας (1 μικρό)", 200));
        db.addBradyna(new Bradyna("Ζελέ χαμηλών θερμίδων (1 μπολ
220γρ.)", 9));
        db.addBradyna(new Bradyna("Ζαμπόν διαίτης (100 γρ.)", 165));
        db.addBradyna(new Bradyna("Ηλιόσποροι (ακαθάριστοι 100 γρ.)",
302));
        db.addBradyna(new Bradyna("Κεράσια (1 κιλό)", 630));
        db.addBradyna(new Bradyna("Κεράσια τάρτα με κρέμα (μερίδα
60γρ.)", 180));
        db.addBradyna(new Bradyna("Κολοκυθάκια βραστά σαλάτα (μερίδα
200γρ.)",
180));
        db.addBradyna(new Bradyna("Κάστανα βραστά ή ψητά (2)", 110));
        db.addBradyna(new Bradyna("Κάσιους (100γρ.)", 600));
        db.addBradyna(new Bradyna("Κασέρι τυρί (100 γρ.)", 390));
        db.addBradyna(new Bradyna("Κορόμηλα κομπόστα (3)", 135));
        db.addBradyna(new Bradyna(
"Κορν φλέικς (1 φλ. τσαγιού) + Γάλα αγελάδος
εβαπορέ (1/2 φλ. τσαγιού)",

```

```

        300));
        db.addBradyna(new Bradyna("Λέμονπάι", 400));
        db.addBradyna(new Bradyna("Λουκουμάδες", 310));
        db.addBradyna(new Bradyna("Μανταρίνια (5)", 225));
        db.addBradyna(new Bradyna(
            "Μούσλι (50 γρ.) + Γάλα αγελάδος εβαπορέ (1/2 φλ.
τσαγιού)",
            405));
        db.addBradyna(new Bradyna("Μουστοκούλουρο (1)", 140));
        db.addBradyna(new Bradyna("Μπισκότα σοκολάτας (1 διπλό)", 165));
        db.addBradyna(new Bradyna("Μπισκότα ginger (3 μικρά)", 55));
        db.addBradyna(new Bradyna("Μπισκότα σοκολάτας cookies με
φιστίκια (2)",
            220));
        db.addBradyna(new Bradyna("Μπρόκολο βραστό σαλάτα", 50));
        db.addBradyna(new Bradyna("Μπισκότα απλά (1)", 230));
        db.addBradyna(new Bradyna("Μπουγάτσα", 360));
        db.addBradyna(new Bradyna("Μήλα (1 φιρίκι)", 65));
        db.addBradyna(new Bradyna("Προφιτερόλ (1 μπολ)", 540));
        db.addBradyna(new Bradyna("Πορτοκάλι (1)", 77));
        db.addBradyna(new Bradyna("Ποπκορν (1 φλιτζάνι)", 135));
        db.addBradyna(new Bradyna("Παστέλι (100 γρ.)", 640));
        db.addBradyna(new Bradyna("Πίτσα σπέσιαλ (1 μεγάλο τρίγωνο)",
400));
        db.addBradyna(new Bradyna("Ραβανί", 365));
        db.addBradyna(new Bradyna("Ρυζόγαλο (1 μπολ)", 325));
        db.addBradyna(new Bradyna("Ροδάκινα τάρτα", 395));
        db.addBradyna(new Bradyna("Σοκολατάκια (πολύ μικρά τα 5)",
110));
        db.addBradyna(new Bradyna("Σοκολατίνα πάστα", 610));
        db.addBradyna(new Bradyna("Σοκολατάκια τρούφες (1)", 60));
        db.addBradyna(new Bradyna("Σύκα ξερά (1)", 65));
    }

```

Κώδικας εισαγωγής βραδινών στη βάση δεδομένων.

8.1.14 FindDietActivity.java

```

package com.example.dietdroid;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;

```



```

import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnTouchListener;
import android.view.animation.AnimationUtils;
import android.view.inputmethod.InputMethodManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageSwitcher;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TabHost;
import android.widget.TabHost.OnTabChangeListener;
import android.widget.TabHost.TabSpec;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewSwitcher;
import android.widget.ViewSwitcher.ViewFactory;

public class FindDietActivity extends Activity implements ViewFactory {
    TabHost host;
    ImageSwitcher imsw1;
    String weight = "...", height = "...", age = "...",
        activity_at_work = "...", activity_outof_work = "...",
sex = "...",
        bmi = "...";
    ArrayAdapter<String> adapter_2, adapter_3;
    Spinner spin_1, spin_2, spin_4;
    EditText editT_1, editT_2, editT_3, editT_4, editT_5, editT_6,
editT_7,
        editT_8;
    TextView textV_11, textV_12;
    ViewSwitcher viewswitcher_1, viewswitcher_2;
    TabSpec inputTab, calcuTab;
    ImageButton imgbut_1;

```

```

int[] imageResources = { R.drawable.question, R.drawable.male,
                        R.drawable.female }, imageResources_without_hint = {
                        R.drawable.male, R.drawable.female };
int curIndex = 0, downX, upX;
SharedPreferences sharedPreferences;
static float draft_5;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    new FullScreen(this);
    setContentView(R.layout.finddietlayout);
    sharedPreferences = PreferenceManager
        .getDefaultSharedPreferences(this);

    // ////////////////////////////////////////
    // OTI AFORA TO TABHOST ///
    host = (TabHost) findViewById(R.id.tabhost1);
    host.setup();
    // kanei setup to tabhost mas , wste na arxikopoihthei
    // ////////////////////////////////////////
    // TO 1o TAB ////////////////////////////////////////
    inputTab = host.newTabSpec("input_tab");
    // anafora sto 1o tab mas
    inputTab.setIndicator(
        getResources().getString(R.string.firsttab_title),
getResources().getDrawable(R.drawable.keyboard_icon));
    // orizw titlo kai icon tou 1o tab
    inputTab.setContent(R.id.include1);
    // orizw pou tha topothetisw ta dedomena tou 1o tab
    host.addTab(inputTab);
    // prosthetw to 1o tab sto tabhost
    // ////////////////////////////////////////
    // TO 2o TAB ////////////////////////////////////////
    calcuTab = host.newTabSpec("calculator_tab");
    // anafora sto 2o tab mas
    calcuTab.setIndicator(getResources()
        .getString(R.string.secondtab_title),
getResources()
        .getDrawable(R.drawable.calculator_icon));
    // orizw titlo kai icon tou 2o tab

```

```

        calcuTab.setContent(R.id.include2);
        // orizw pou tha topothetisw ta dedomena tou 2o tab
        host.addTab(calcuTab);
        // prosthetw to 1o tab sto tabhost
        // //////////////////////////////////////
        for (int i = 0; i < host.getTabWidget().getChildCount(); i++) {

            host.getTabWidget().getChildAt(i).getLayoutParams().height *= 1.2;
            host.getTabWidget().getChildAt(i)

                .setBackgroundResource(R.drawable.tab_background_selectors);
            // orizw me ayto ton tropo tis diafores states tou
background tou tab

                TextView tab_text = (TextView)
host.getTabWidget().getChildAt(i)

                    .findViewById(android.R.id.title);
                tab_text.setTextColor(R.drawable.tab_text_selectors);
                /*---- den xrisimopoihthike o tropos aytos kathws
epitrepei mono sugkekrimeno xrwma
                ---- xrisimopoihthike to style gia na kathorisoume ta
states tou textColor*/
                }
            // allazw kapoia features gia na fainontai icon & text sta tabs
(ayta se
            // syndiasmo me ayto sto manifest)

            host.getTabWidget().setLeftStripDrawable(R.drawable.tab_strip_selector
);
            host.getTabWidget()

                .setRightStripDrawable(R.drawable.tab_strip_selector);
            // orizw ti tha xrisimopieitai gia strip twv tabs
            host.setCurrentTabByTag("input_tab");
            // orizw pio einai arxika epilegmeno

            editT_4 = (EditText) findViewById(R.id.editT4);
            editT_5 = (EditText) findViewById(R.id.editT5);
            editT_6 = (EditText) findViewById(R.id.editT6);
            editT_7 = (EditText) findViewById(R.id.editT7);
            editT_8 = (EditText) findViewById(R.id.editT8);

            imgbut_1 = (ImageButton) findViewById(R.id.imgbut1);
            imgbut_1.setVisibility(View.GONE);

```

```

        host.setOnTabChangeListener(new OnTabChangeListener() {

            @Override
            public void onTabChanged(String tabId) {

                if (host.getCurrentTab() == 1) {
                    // // xrisimopoioume auton ton kwdika gia na
min uparxei i
                    // periptwsi
                    // "otan allazw tabs me ανοικτο keyboard na
min fainetai sto calculatb keyboard"
                    // xrisimopoiw to editT_4 giati einai to 1
sto calculatb
                    InputMethodManager imm =
(InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);

                    imm.hideSoftInputFromWindow(editT_4.getWindowToken(), 0);

                    // / kanei ena toast short message me tis
times pou exoume
                    // balei////
                    Toast.makeText(
                        getApplicationContext(),
                        "Your Weight = " + weight +
                        + height + "\n" +
                        "Your Age = " + age
                        + "\n" + "Your Sex
                        = " + sex + "\n"
                        + "Your physical
                        activity at work = "
                        + activity_at_work
                        + "\n"
                        + "Your physical
                        activity out of work = "
                        +
                        activity_outof_work, Toast.LENGTH_SHORT)
                        .show();

                    float draft = 0, draft_cat = 0, draft_2 = 0,
draft_3 = 0, draft_4 = 0;
                    draft_5 = 0;
                    ;

```

```

        if ((weight.equals("...")) &&
(height.equals("..."))) {
            editT_4.setText("");

            editT_4.setHint(R.string.calcu_hint_1);
            editT_5.setText("");

            editT_5.setHint(R.string.calcu_hint_1);
            } else if ((weight.equals("..."))
                && (!(height.equals("...")))) {
                editT_4.setText("");

            editT_4.setHint(R.string.calcu_hint_2);
            editT_5.setText("");

            editT_5.setHint(R.string.calcu_hint_2);
            } else if (!(weight.equals("..."))
                && (height.equals("..."))) {
                editT_4.setText("");

            editT_4.setHint(R.string.calcu_hint_3);
            editT_5.setText("");

            editT_5.setHint(R.string.calcu_hint_3);
            } else {
                draft = (Float.parseFloat(weight) *
10000 / ((Float
                .parseFloat(height)) *
(Float
                .parseFloat(height))));

                if (draft < 18.5) {

                    editT_5.setText(R.string.elleipobaris);
                    draft_cat = 1;
                } else if ((draft >= 18.5) && (draft <
25)) {

                    editT_5.setText(R.string.fysiologikos);
                    draft_cat = 2;
                } else if ((draft >= 25) && (draft <
30)) {

                    editT_5.setText(R.string.yperbaros);
                    draft_cat = 3;

```

```

    } else if ((draft >= 30) && (draft <
35)) {
        editT_5.setText(R.string.first_bathmou_yperbaros);
        draft_cat = 4;
    } else if ((draft >= 35) && (draft <
40)) {
        editT_5.setText(R.string.second_bathmou_yperbaros);
        draft_cat = 5;
    } else if (draft >= 40) {
        editT_5.setText(R.string.pathologika_yperbaros);
        draft_cat = 6;
    }
    String string_draft =
String.valueOf(draft);
    string_draft = String.format("%.2f",
draft);
    // / to xrisimopoiume san format tou
string gia na mas
    // dinei mono 2 demicals psifia
    editT_4.setText(string_draft);
}
if ((sex.equals("...")) && (draft == 0)
    && (age.equals("..."))) {
    editT_6.setText("");
}
editT_6.setHint(R.string.calcu_hint_4);// 000
} else if (!(sex.equals("..."))) && (draft
== 0)
    && (age.equals("..."))) {
    editT_6.setText("");
}
editT_6.setHint(R.string.calcu_hint_5);// x00
} else if ((sex.equals("...")) && (draft !=
0)
    && (age.equals("..."))) {
    editT_6.setText("");
}
editT_6.setHint(R.string.calcu_hint_6);// 0x0
} else if ((sex.equals("...")) && (draft ==
0)
    && (!(age.equals("...")))) {
    editT_6.setText("");
}

```

```

editT_6.setHint(R.string.calcu_hint_7);// 00x
    } else if (!(sex.equals("..."))) && (draft
!= 0)
        && (age.equals("..."))) {
            editT_6.setText("");

editT_6.setHint(R.string.calcu_hint_8);// xx0
    } else if (!(sex.equals("..."))) && (draft
== 0)
        && (!(age.equals("...")))) {
            editT_6.setText("");

editT_6.setHint(R.string.calcu_hint_9);// x0x
    } else if ((sex.equals("...")) && (draft !=
0)
        && (!(age.equals("...")))) {
            editT_6.setText("");

editT_6.setHint(R.string.calcu_hint_10);// 0xx
    } else {
        float idaniko_baros = (float) (22.7 *
(Float
        .parseFloat(height)) *
(Float
        .parseFloat(height))) /
10000;
        float prosarmosmeno_baros = (float)
(((Float
        .parseFloat(weight)) -
idaniko_baros) * (0.25) + idaniko_baros);
        if (draft_cat == 1) {
            if (sex.equals("male")) {
                draft_3 =
calculate_male_bmr(draft_2,
                idaniko_baros);
            } else {
                draft_3 =
calculate_female_bmr(draft_2,
                idaniko_baros);
            }
        } else if ((draft_cat == 2) &&
(draft_cat == 3)) {
            if (sex.equals("male")) {

```

```

                                                                    draft_3 =
calculate_male_bmr(draft_2,
                    Float.parseFloat(weight));
                                                                    } else {
                                                                    draft_3 =
calculate_female_bmr(draft_2,
                    Float.parseFloat(weight));
                                                                    }
                                                                    } else {
                                                                    if (sex.equals("male")) {
                                                                    draft_3 =
calculate_male_bmr(draft_2,
                    prosarmosmeno_baros);
                                                                    } else {
                                                                    draft_3 =
calculate_female_bmr(draft_2,
                    prosarmosmeno_baros);
                                                                    }
                                                                    }
                                                                    }
                                                                    if ((activity_at_work.equals("..."))
                                                                    &&
                                                                    (activity_outof_work.equals("..."))
                                                                    && (sex.equals("..."))) {
                                                                    editT_7.setText("");

                                                                    editT_7.setHint(R.string.calcu_hint_11);// 000
                                                                    } else if
                                                                    ((!(activity_at_work.equals("...")))
                                                                    &&
                                                                    (activity_outof_work.equals("..."))
                                                                    && (sex.equals("..."))) {
                                                                    editT_7.setText("");

                                                                    editT_7.setHint(R.string.calcu_hint_11);// x00
                                                                    } else if ((activity_at_work.equals("..."))
                                                                    &&
                                                                    (!(activity_outof_work.equals("...")))
                                                                    && (sex.equals("..."))) {
                                                                    editT_7.setText("");

```



```

        editT_7.setHint(R.string.calcu_hint_11);// 0x0
        } else if ((activity_at_work.equals("..."))
                &&
(activity_outof_work.equals("..."))
                && (!(sex.equals("...")))) {
            editT_7.setText("");

        editT_7.setHint(R.string.calcu_hint_12);// 00x
        } else if
((!(activity_at_work.equals("...")))
                &&
(!(activity_outof_work.equals("...")))
                && (sex.equals("..."))) {
            editT_7.setText("");

        editT_7.setHint(R.string.calcu_hint_10);// xx0
        } else if
((!(activity_at_work.equals("...")))
                &&
(activity_outof_work.equals("..."))
                && (!(sex.equals("...")))) {
            editT_7.setText("");

        editT_7.setHint(R.string.calcu_hint_12);// x0x
        } else if ((activity_at_work.equals("..."))
                &&
(!(activity_outof_work.equals("...")))
                && (!(sex.equals("...")))) {
            editT_7.setText("");

        editT_7.setHint(R.string.calcu_hint_12);// 0xx
        } else {
            if
((activity_outof_work.equals("light"))
                &&
(activity_at_work.equals("light"))
                && (sex.equals("male")))
        {
            draft_4 = (float) 0.4;
            editT_7.setText("" + draft_4);
        } else if
((activity_outof_work.equals("light"))
                &&
(activity_at_work.equals("light")))

```

```

                                                                 &&
((sex.equals("female"))) {
    draft_4 = (float) 0.4;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("light"))
                                                                 &&
(activity_at_work.equals("normal"))
                                                                 && (sex.equals("male")))
{
    draft_4 = (float) 0.6;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("light"))
                                                                 &&
(activity_at_work.equals("normal"))
                                                                 &&
(sex.equals("female"))) {
    draft_4 = (float) 0.5;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("light"))
                                                                 &&
(activity_at_work.equals("intensive"))
                                                                 && (sex.equals("male")))
{
    draft_4 = (float) 0.7;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("light"))
                                                                 &&
(activity_at_work.equals("intensive"))
                                                                 &&
(sex.equals("female"))) {
    draft_4 = (float) 0.5;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("normal"))
                                                                 &&
(activity_at_work.equals("light"))
                                                                 && (sex.equals("male")))
{
    draft_4 = (float) 0.5;
    editT_7.setText("" + draft_4);
} else if
((activity_outof_work.equals("normal"))

```

```

                                                                    &&
(activity_at_work.equals("light"))
                                                                    &&
(sex.equals("female"))) {
                                                                    draft_4 = (float) 0.5;
                                                                    editT_7.setText("" + draft_4);
                                                                    } else if
((activity_outof_work.equals("normal"))
                                                                    &&
(activity_at_work.equals("normal"))
                                                                    && (sex.equals("male")))
{
                                                                    draft_4 = (float) 0.7;
                                                                    editT_7.setText("" + draft_4);
                                                                    } else if
((activity_outof_work.equals("normal"))
                                                                    &&
(activity_at_work.equals("normal"))
                                                                    &&
(sex.equals("female"))) {
                                                                    draft_4 = (float) 0.6;
                                                                    editT_7.setText("" + draft_4);
                                                                    } else if
((activity_outof_work.equals("normal"))
                                                                    &&
(activity_at_work.equals("intensive"))
                                                                    && (sex.equals("male")))
{
                                                                    draft_4 = (float) 0.8;
                                                                    editT_7.setText("" + draft_4);
                                                                    } else if
((activity_outof_work.equals("normal"))
                                                                    &&
(activity_at_work.equals("intensive"))
                                                                    &&
(sex.equals("female"))) {
                                                                    draft_4 = (float) 0.6;
                                                                    editT_7.setText("" + draft_4);
                                                                    } else if
((activity_outof_work.equals("intensive"))
                                                                    &&
(activity_at_work.equals("light"))
                                                                    && (sex.equals("male")))
{
                                                                    draft_4 = (float) 0.6;
                                                                    editT_7.setText("" + draft_4);

```

```

        } else if
        ((activity_outof_work.equals("intensive"))
            &&
            (activity_at_work.equals("light"))
            &&
            (sex.equals("female"))) {
                draft_4 = (float) 0.6;
                editT_7.setText("" + draft_4);
        } else if
        ((activity_outof_work.equals("intensive"))
            &&
            (activity_at_work.equals("normal"))
            && (sex.equals("male")))
        {
                draft_4 = (float) 0.8;
                editT_7.setText("" + draft_4);
        } else if
        ((activity_outof_work.equals("intensive"))
            &&
            (activity_at_work.equals("normal"))
            &&
            (sex.equals("female"))) {
                draft_4 = (float) 0.7;
                editT_7.setText("" + draft_4);
        } else if
        ((activity_outof_work.equals("intensive"))
            &&
            (activity_at_work.equals("intensive"))
            && (sex.equals("male")))
        {
                draft_4 = (float) 0.9;
                editT_7.setText("" + draft_4);
        } else if
        ((activity_outof_work.equals("intensive"))
            &&
            (activity_at_work.equals("intensive"))
            &&
            (sex.equals("female"))) {
                draft_4 = (float) 0.7;
                editT_7.setText("" + draft_4);
        }
    }
    if ((draft_3 == 0) && (draft_4 == 0)) {
        editT_8.setText("");
    }

```

```

        editT_8.setHint(R.string.calcu_hint_13);
        } else if ((draft_3 != 0) && (draft_4 == 0))
    {
        editT_8.setText("");

        editT_8.setHint(R.string.calcu_hint_14);
        } else if ((draft_3 == 0) && (draft_4 != 0))
    {
        editT_8.setText("");

        editT_8.setHint(R.string.calcu_hint_15);
        } else {
draft_4);
draft_5 = draft_3 + (draft_3 *
String.valueOf(draft_5);
String string_draft_5 =
draft_5);
string_draft_5 = String.format("%.2f",
string gia na mas
// / to xrisimopoiame san format tou
// dinei mono 2 demicals psifia
editT_8.setText(string_draft_5);
imgbut_1.setClickable(true);
imgbut_1.setVisibility(View.VISIBLE);
// emfanise to icon_2 otan
symplrwthoun ola , grafetee dw giati na na deixoume to edit_8 simainei oti
exoun sumplrwthei ola
        }
        if ((activity_at_work.equals("..."))
            ||
(activity_outof_work.equals("..."))
            || (weight.equals("...")) ||
(height.equals("..."))
            || (sex.equals("...")) ||
(age.equals("..."))) {
            imgbut_1.setClickable(false);
            imgbut_1.setVisibility(View.GONE);
        }
        //an svisoume kapoio stoixeio xanakrybei to
icon_2
        }
    }
});

```

```

// ////////////////////////////////////////
// OTI AFORA TA EDITTEXT////
editT_1 = (EditText) findViewById(R.id.editT1);
editT_1.addTextChangedListener(new TextWatcher() {

    @Override
    public void afterTextChanged(Editable arg0) {
        weight = editT_1.getText().toString();
        if (weight.equals("")) {
            weight = "...";
        } else if ((Integer.parseInt(weight)) >
Integer.parseInt(sharedPreferences.getString("max_weight", "default"))) {
            weight =
sharedPreferences.getString("max_weight", "default");
            editT_1.setText(weight);
        } else if (arg0.charAt(0) == '0') {
            weight = "...";
            editT_1.setText("");
        }
        // gia na min mporei na baleis arithmous tupou 000
, 001 , 011

        editT_1.setSelection(editT_1.length());
        // gia na paei o cursor meta tin metatropi

    }

    @Override
    public void beforeTextChanged(CharSequence s, int start,
int count,
        int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int
before,
        int count) {

    }

});
editT_2 = (EditText) findViewById(R.id.editT2);
editT_2.addTextChangedListener(new TextWatcher() {

```

```

        @Override
        public void afterTextChanged(Editable arg0) {
            height = editT_2.getText().toString();
            if (height.equals("")) {
                height = "...";
            } else if ((Integer.parseInt(height)) >
Integer.parseInt(sharedPreferences.getString("max_height", "default"))) {
                height =
sharedPreferences.getString("max_height", "default");
                editT_2.setText(height);
            } else if (arg0.charAt(0) == '0') {
                height = "...";
                editT_2.setText("");
            }
            // gia na min mporei na baleis arithmous tupou 000
, 001 , 011

            editT_2.setSelection(editT_2.length());
            // gia na paei o cursor meta tin metatropi

        }

        @Override
        public void beforeTextChanged(CharSequence s, int start,
int count,
            int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int
before,
            int count) {

        }

    });
    editT_3 = (EditText) findViewById(R.id.editT3);
    editT_3.addTextChangedListener(new TextWatcher() {

        @Override
        public void afterTextChanged(Editable arg0) {
            age = editT_3.getText().toString();

```

```

        if (age.equals("")) {
            age = "...";
        } else if ((Integer.parseInt(age)) >
Integer.parseInt(sharedPreferences.getString("max_age", "default"))) {
            age = sharedPreferences.getString("max_age",
"default");

            editT_3.setText(age);
        } else if (arg0.charAt(0) == '0') {
            age = "...";
            editT_3.setText("");
        }
        // gia na min mporei na baleis arithmous tupou 000
, 001 , 011

        editT_3.setSelection(editT_3.length());
        // gia na paei o cursor meta tin metatropi
    }

    @Override
    public void beforeTextChanged(CharSequence s, int start,
int count,
        int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int
before,
        int count) {

    }

});
// ////////////////////////////////////////
// OTI AFORA TON IMAGESLIDER///
imsw1 = (ImageSwitcher) findViewById(R.id.imsw1);
imsw1.setFactory(this);
imsw1.setInAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_in));
imsw1.setOutAnimation(AnimationUtils.loadAnimation(this,
        android.R.anim.fade_out));
imsw1.setImageResource(imageResources[curIndex]);
imsw1.setOnTouchListener(new OnTouchListener() {

```



```

@Override
public boolean onTouch(View v, MotionEvent event) {

    imsw1.setImageResource(imageResources_without_hint[curIndex]);
    // /afou kaneis klik bgazeis tin prwti fwto pou
    einai to hint
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        downX = (int) event.getX();
        sex = Integer.toString(curIndex);
        if (sex.equals("0")) {
            sex = "male";
        } else if (sex.equals("1")) {
            sex = "female";
        }
        return true;
    } else if (event.getAction() ==
MotionEvent.ACTION_UP) {
        upX = (int) event.getX();
        if (upX - downX > 1) {
            curIndex--;
            if (curIndex < 0) {
                curIndex =
imageResources_without_hint.length - 1;
            }

            imsw1.setInAnimation(AnimationUtils.loadAnimation(
                getApplicationContext(),
                android.R.anim.slide_in_left));

            imsw1.setOutAnimation(AnimationUtils.loadAnimation(
                getApplicationContext(),
                android.R.anim.slide_out_right));

            imsw1.setImageResource(imageResources_without_hint[curIndex]);
            } else if (downX - upX > -1) {
                curIndex++;
                if (curIndex >
imageResources_without_hint.length - 1) {
                    curIndex = 0;
                }
            }
        }
    }
}

```

```

        imsw1.setInAnimation(AnimationUtils.loadAnimation(
                                                    getApplicationContext(),
        android.R.anim.slide_in_left));

        imsw1.setOutAnimation(AnimationUtils.loadAnimation(
                                                    getApplicationContext(),
        android.R.anim.slide_out_right));

        imsw1.setImageResource(imageResources_without_hint[curIndex]);
        }
        sex = Integer.toString(curIndex);
        if (sex.equals("0")) {
            sex = "male";
        } else if (sex.equals("1")) {
            sex = "female";
        }
        return true;
    }
    return false;
}

});

// ////////////////////////////////////////
// OTI AFORA TOUS SPINNERS ///

        adapter_2 = new ArrayAdapter<String>(FindDietActivity.this,
        R.layout.spinner_item,
        getResources().getStringArray(
        R.array.eidh_drastiriothtw));
        spin_1 = (Spinner) findViewById(R.id.spin1);
        spin_1.setAdapter(adapter_2);
        textV_11 = (TextView) findViewById(R.id.textV11);
        viewswitcher_1 = (ViewSwitcher)
        findViewById(R.id.viewswitcher1);
        textV_11.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                viewswitcher_1.showNext();
            }
        });
    }
}

```

```

        spin_1.performClick();
        activity_at_work = getResources().getStringArray(
            R.array.eidh_drastiriothtw)[spin_1
                .getSelectedItemPosition()];
    }

});
spin_1.setOnItemClickListener(new OnItemSelectedListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View
arg1,
        int arg2, long arg3) {
        activity_at_work = getResources().getStringArray(
            R.array.eidh_drastiriothtw)[spin_1
                .getSelectedItemPosition()];

    }

    @Override
    public void onNothingSelected(AdapterView<?> arg0) {

    }

});
adapter_3 = new ArrayAdapter<String>(FindDietActivity.this,
    R.layout.spinner_item,
getResources().getStringArray(
        R.array.eidh_drastiriothtw));
spin_2 = (Spinner) findViewById(R.id.spin2);
spin_2.setAdapter(adapter_3);
viewswitcher_2 = (ViewSwitcher)
findViewById(R.id.viewswitcher2);
textV_12 = (TextView) findViewById(R.id.textV12);
textV_12.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        spin_2.performClick();
        viewswitcher_2.showNext();

        activity_outof_work =
getResources().getStringArray(
            R.array.eidh_drastiriothtw)[spin_2

```

```

        .getSelectedItemPosition());
    }
});
spin_2.setOnItemClickListener(new OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> arg0, View
arg1,
        int arg2, long arg3) {
        activity_outof_work =
getResources().getStringArray(
        R.array.eidh_drastiriothtw)[spin_2
        .getSelectedItemPosition()];
    }

    @Override
    public void onNothingSelected(AdapterView<?> arg0) {

    }
});
// ////////////////////////////////////////
// OTI AFORA TO IMAGEBUTTON - calcul //
imgbut_1.setOnClickListener(new OnClickListener(){

    @Override
    public void onClick(View arg0) {
        // Launch the Settings Activity
        startActivity(new Intent(FindDietActivity.this,
ResultsActivity.class));
    }

});

}

@Override
public View makeView() {
    ImageView i = new ImageView(this);
    i.setScaleType(ImageView.ScaleType.FIT_CENTER);
    return i;
}

public float calculate_male_bmr(float dr_2, float bar) {

```

```

        dr_2 = (float) (66.473 + (13.7516 * bar)
            + (5.0033 * (Float.parseFloat(height)))) - (6.755 *
(Float
            .parseFloat(age)));
        String string_draft_2 = String.valueOf(dr_2);
        string_draft_2 = String.format("%.2f", dr_2);
        // / to xrisimopoiume san format tou string gia na mas
        // dinei mono 2 demicals psifia
        editT_6.setText(string_draft_2);
        return dr_2;
    }

    public float calculate_female_bmr(float dr_2, float bar) {
        dr_2 = (float) (655.0955 + (9.5634 * bar)
            + (1.8496 * (Float.parseFloat(height)))) - (4.6756 *
(Float
            .parseFloat(age)));
        String string_draft_2 = String.valueOf(dr_2);
        string_draft_2 = String.format("%.2f", dr_2);
        // / to xrisimopoiume san format tou string gia na mas
        // dinei mono 2 demicals psifia
        editT_6.setText(string_draft_2);
        return dr_2;
    }
}

```

8.1.15 ResultsActivity.java

```

package com.example.dietdroid;

import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.List;
import java.util.Random;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

```

```
import android.provider.CalendarContract;
import android.provider.CalendarContract.Events;
import android.util.Log;
import android.view.View;
import android.view.View.OnLongClickListener;
import android.widget.TabHost;
import android.widget.Toast;
import android.widget.TabHost.OnTabChangeListener;
import android.widget.TabHost.TabSpec;
import android.widget.TextView;

public class ResultsActivity extends Activity {

    TabHost host_2;
    TabSpec monTab, tueTab, wenTab, thurTab, friTab, satTab, sunTab;
    DatabaseHandler db;
    TextView textV_24, textV_25, textV_26, textV_27, textV_28, textV_29,
        textV_30, textV_31;

    int random_prwino, random_mesimeriano, random_bradyo, sum_calories;
    List<Prwina> prwina;
    List<Mesimeriana> mesimeriana;
    List<Bradyna> bradyo;
    View tabView_1, tabView_2, tabView_3, tabView_4, tabView_5, tabView_6,
        tabView_7;
    Calendar calendar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

        super.onCreate(savedInstanceState);
        new FullScreen(this);
        setContentView(R.layout.resultslayout);

        textV_24 = (TextView) findViewById(R.id.textV24);
        textV_25 = (TextView) findViewById(R.id.textV25);
        textV_26 = (TextView) findViewById(R.id.textV26);
        textV_27 = (TextView) findViewById(R.id.textV27);
        textV_28 = (TextView) findViewById(R.id.textV28);
        textV_29 = (TextView) findViewById(R.id.textV29);
        textV_30 = (TextView) findViewById(R.id.textV30);
        textV_31 = (TextView) findViewById(R.id.textV31);

        textV_24.setText("Breakfast (<="
            + (int) (0.3 * 0.9 * FindDietActivity.draft_5) +
        "));

        textV_26.setText("Lunch (<="
            + (int) (0.5 * 0.9 * FindDietActivity.draft_5) +
        "));

        textV_28.setText("Dinner (<="
            + (int) (0.2 * 0.9 * FindDietActivity.draft_5) +
        "));

        textV_30.setText("Total Calories");

        db = new DatabaseHandler(this);
        prwina = db.getPrwina();
        mesimeriana = db.getMesimeriana();
        bradyna = db.getBradyna();

        // //////////////////////////////////////

```

```
// OTI AFORA TO TABHOST ///
```

```
host_2 = (TabHost) findViewById(R.id.tabhost2);
```

```
host_2.setup();
```

```
// kanei setup to tabhost mas , wste na arxikopoihthei
```

```
// //////////////////////////////////////
```

```
// //////////////////////////////////////
```

```
// //////////////////////////////////////
```

```
monTab = host_2.newTabSpec("mon_tab");
```

```
// anafora sto 1o tab mas
```

```
monTab.setIndicator(getResources().getStringArray(
```

```
                R.array.imeres_ebdomadas)[0]);
```

```
// orizw titlo kai icon tou 1o tab
```

```
monTab.setContent(R.id.include3);
```

```
// orizw pou tha topothetisw ta dedomena tou 1o tab
```

```
host_2.addTab(monTab);
```

```
// prosthetw to 1o tab sto tabhost
```

```
// //////////////////////////////////////
```

```
// //////////////////////////////////////
```

```
// //////////////////////////////////////
```

```
tueTab = host_2.newTabSpec("tue_tab");
```

```
// anafora sto 2o tab mas
```

```
tueTab.setIndicator(getResources().getStringArray(
```

```
                R.array.imeres_ebdomadas)[1]);
```

```
// orizw titlo kai icon tou 2o tab
```

```
tueTab.setContent(R.id.include3);
```

```
// orizw pou tha topothetisw ta dedomena tou 2o tab
```

```
host_2.addTab(tueTab);
```

```
// prosthetw to 2o tab sto tabhost
```

```
// //////////////////////////////////////
```



```
// ////////// TO 3o TAB ////////////////////////////////////////
wenTab = host_2.newTabSpec("wen_tab");
// αναφορά στο 3ο tab mas
wenTab.setIndicator(getResources().getStringArray(
    R.array.imeres_ebdomadas)[2]);
// οριζω τίτλο και icon του 3ο tab
wenTab.setContent(R.id.include3);
// οριζω που θα τοποθετησw τα δεδομένα του 3ο tab
host_2.addTab(wenTab);
// προσθηtw to 3ο tab sto tabhost
// ////////////////////////////////////////

// ////////// TO 4o TAB ////////////////////////////////////////
thurTab = host_2.newTabSpec("thu_tab");
// αναφορά στο 4ο tab mas
thurTab.setIndicator(getResources().getStringArray(
    R.array.imeres_ebdomadas)[3]);
// οριζω τίτλο και icon του 4ο tab
thurTab.setContent(R.id.include3);
// οριζω που θα τοποθετησw τα δεδομένα του 4ο tab
host_2.addTab(thurTab);
// προσθηtw to 4ο tab sto tabhost
// ////////////////////////////////////////

// ////////// TO 5o TAB ////////////////////////////////////////
friTab = host_2.newTabSpec("fri_tab");
// αναφορά στο 5ο tab mas
friTab.setIndicator(getResources().getStringArray(
    R.array.imeres_ebdomadas)[4]);
```

```
// orizw titlo kai icon tou 5o tab
friTab.setContent(R.id.include3);

// orizw pou tha topothetisw ta dedomena tou 5o tab
host_2.addTab(friTab);

// prosthetw to 5o tab sto tabhost
// //////////////////////////////////////

// ===== TO 6o TAB =====
satTab = host_2.newTabSpec("sat_tab");

// anafora sto 6o tab mas
satTab.setIndicator(getResources().getStringArray(
    R.array.imeres_ebdomadas)[5]);

// orizw titlo kai icon tou 6o tab
satTab.setContent(R.id.include3);

// orizw pou tha topothetisw ta dedomena tou 6o tab
host_2.addTab(satTab);

// prosthetw to 6o tab sto tabhost
// //////////////////////////////////////

// ===== TO 7o TAB =====
sunTab = host_2.newTabSpec("sun_tab");

// anafora sto 7o tab mas
sunTab.setIndicator(getResources().getStringArray(
    R.array.imeres_ebdomadas)[6]);

// orizw titlo kai icon tou 7o tab
sunTab.setContent(R.id.include3);

// orizw pou tha topothetisw ta dedomena tou 7o tab
host_2.addTab(sunTab);

// prosthetw to 7o tab sto tabhost
```

```

// ////////////////////////////////////////

for (int i = 0; i < host_2.getTabWidget().getChildCount(); i++)
{

    host_2.getTabWidget().getChildAt(i).getLayoutParams().height *= 1.2;
    host_2.getTabWidget().getChildAt(i)

        .setBackgroundResource(R.drawable.tab_background_selectors);
    // orizw me ayto ton tropo tis diafores states tou
background tou tab

        TextView tab_text = (TextView)
host_2.getTabWidget().getChildAt(i)

            .findViewById(android.R.id.title);
        tab_text.setTextColor(R.drawable.tab_text_selectors);
        /*---- den xrisimopoihthike o tropos aytos kathws
epitrepei mono sugkekrimeno xrwma

        ---- xrisimopoihthike to style gia na kathorisoume ta
states tou textColor*/

    }

    // allazw kapoia features gia na fainontai icon & text sta tabs
(ayta se

    // syndiasmo me ayto sto manifest)

    host_2.getTabWidget().setLeftStripDrawable(
        R.drawable.tab_strip_selector);
    host_2.getTabWidget().setRightStripDrawable(
        R.drawable.tab_strip_selector);

    // orizw ti tha xrisimopieitai gia strip twv tabs

```

```
        host_2.setCurrentTabByTag("tue_tab");
        // orizw pio einai arxika epilegmeno
        host_2.setCurrentTabByTag("mon_tab");
        // / bazeis current to 2o kai meta to prwto gia na diorthwseis
to bug
        // pou den deixnei to layout sto 1o..
        // (to bug emfanizetai otan bazeis arxika to prwto kai
diorthwnei
        // automata otan pas sto 2o)

        if (prwina.isEmpty()) {
            textV_25.setText("-");
            textV_27.setText("")
                +
mesimeriana.get(random_mesimeriano).getNameMesimeriana());
            textV_29.setText("") +
bradyna.get(random_bradyno).getNameBradyna());
            textV_31.setText("-");
            // an einai i lista me ta prwina adeia kai den den mporei
na parei timi
        } else if (mesimeriana.isEmpty()) {
            textV_25.setText("") +
prwina.get(random_prwino).getNamePrwino());
            textV_27.setText("-");
            textV_29.setText("") +
bradyna.get(random_bradyno).getNameBradyna());
            textV_31.setText("-");
            // an einai i lista me ta mesimeriana adeia kai den den
mporei na parei timi
        } else if (bradyna.isEmpty()) {
            textV_25.setText("") +
prwina.get(random_prwino).getNamePrwino());
            textV_27.setText("")
```

```

+
mesimeriana.get(random_mesimeriano).getNameMesimeriana());
    textView_29.setText("-");
    textView_31.setText("-");
    // an einai i lista me ta bradyna adeia kai den den
mporei na parei timi
    } else {
        random_prwino = new Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
Random().nextInt(mesimeriana.size());
        Log.i("rand", "" + random_mesimeriano);
        random_bradyno = new Random().nextInt(bradyna.size());
        Log.i("rand", "" + random_bradyno);

        textView_25.setText("" +
prwina.get(random_prwino).getNamePrwina());
        textView_27.setText(""
+
mesimeriana.get(random_mesimeriano).getNameMesimeriana());
        textView_29.setText("" +
bradyna.get(random_bradyno).getNameBradyna());
        sum_calories =
prwina.get(random_prwino).getThermidesPrwina()
+ mesimeriana.get(random_mesimeriano)
.getThermidesMesimeriana()
+
bradyna.get(random_bradyno).getThermidesBradyna());
        textView_31.setText("" + sum_calories);

        Toast.makeText(this,
        "Long Press Day' s Name to add event to Calendar",
Toast.LENGTH_LONG).show();

```

```
host_2.setOnTabChangeListener(new OnTabChangeListener()
{
    @Override
    public void onTabChanged(String tabId) {
        // TODO Auto-generated method stub
        if (host_2.getCurrentTab() == 0) {
            random_prwino = new
Random().nextInt(prwina.size());
            Log.i("rand", "" + random_prwino);
            random_mesimeriano = new
Random().nextInt(mesimeriana
.size());
            Log.i("rand", "" +
random_mesimeriano);
            random_bradyo = new
Random().nextInt(bradyo.size());
            Log.i("rand", "" + random_bradyo);
            textV_25.setText(""
+
prwina.get(random_prwino).getNamePrwino());
            textV_27.setText(""
+
mesimeriana.get(random_mesimeriano)
.getNameMesimeriana());
            textV_29.setText(""
+
bradyo.get(random_bradyo).getNameBradyo());
            sum_calories =
prwina.get(random_prwino)
```

```

        .getThermidesPrwina()
        +
mesimeriana.get(random_mesimeriano)

        .getThermidesMesimeriana()
        +
bradyna.get(random_bradyno)

        .getThermidesBradyna());

        textView_31.setText("" + sum_calories);

    }
    if (host_2.getCurrentTab() == 1) {
        random_prwino = new
Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
Random().nextInt(mesimeriana
        .size());
        Log.i("rand", "" +
random_mesimeriano);
        random_bradyno = new
Random().nextInt(bradyna.size());
        Log.i("rand", "" + random_bradyno);

        textView_25.setText(""
        +
prwina.get(random_prwino).getNamePrwina());
        textView_27.setText(""
        +
mesimeriana.get(random_mesimeriano)
        .getNameMesimeriana());
        textView_29.setText(""

```

```

bradyna.get(random_bradyno).getNameBradyna());
+
prwina.get(random_prwino)
sum_calories =
+
.getThermidesPrwina()
+
mesimeriana.get(random_mesimeriano)
+
.getThermidesMesimeriana()
+
bradyna.get(random_bradyno)
+
.getThermidesBradyna());
+
textV_31.setText("" + sum_calories);
}
if (host_2.getCurrentTab() == 2) {
random_prwino = new
Random().nextInt(prwina.size());
Log.i("rand", "" + random_prwino);
random_mesimeriano = new
Random().nextInt(mesimeriana
.size());
Log.i("rand", "" +
random_mesimeriano);
random_bradyno = new
Random().nextInt(bradyna.size());
Log.i("rand", "" + random_bradyno);
textV_25.setText(""
+
prwina.get(random_prwino).getNamePrwina());
textV_27.setText(""
+
mesimeriana.get(random_mesimeriano)

```



```

        .getNameMesimeriana());
                                textView_29.setText(""
                                +
bradyna.get(random_bradyno).getNameBradyna());
                                sum_calories =
prwina.get(random_prwino)
                                .getThermidesPrwino()
                                +
mesimeriana.get(random_mesimeriano)
                                .getThermidesMesimeriana()
                                +
bradyna.get(random_bradyno)
                                .getThermidesBradyna());
                                textView_31.setText("" + sum_calories);
    }
    if (host_2.getCurrentTab() == 3) {
        random_prwino = new
Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
                                .size());
        Log.i("rand", "" +
random_mesimeriano);
        random_bradyno = new
Random().nextInt(bradyna.size());
        Log.i("rand", "" + random_bradyno);

        textView_25.setText(""
                                +
prwina.get(random_prwino).getNamePrwino());

```

```

        textView_27.setText(""
            +
mesimeriana.get(random_mesimeriano)
            +
            .getNameMesimeriana());
        textView_29.setText(""
            +
bradyna.get(random_bradyno).getNameBradyna());
        sum_calories =
prwina.get(random_prwino)
            +
            .getThermidesPrwina()
            +
mesimeriana.get(random_mesimeriano)
            +
            .getThermidesMesimeriana()
            +
bradyna.get(random_bradyno)
            +
            .getThermidesBradyna());
        textView_31.setText("" + sum_calories);
    }
    if (host_2.getCurrentTab() == 4) {
        random_prwino = new
Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
Random().nextInt(mesimeriana
            .size());
        Log.i("rand", "" +
random_mesimeriano);
        random_bradyno = new
Random().nextInt(bradyna.size());
        Log.i("rand", "" + random_bradyno);
    }
}

```

```

        textView_25.setText(""
                                +
prwina.get(random_prwino).getNamePrwina());
        textView_27.setText(""
                                +
mesimeriana.get(random_mesimeriano)
                .getNameMesimeriana());
        textView_29.setText(""
                                +
bradyna.get(random_bradyno).getNameBradyna());
        sum_calories =
prwina.get(random_prwino)
                .getThermidesPrwina()
        +
mesimeriana.get(random_mesimeriano)
                .getThermidesMesimeriana()
        +
bradyna.get(random_bradyno)
                .getThermidesBradyna();
        textView_31.setText("" + sum_calories);
    }
    if (host_2.getCurrentTab() == 5) {
        random_prwino = new
Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
Random().nextInt(mesimeriana
                .size());
        Log.i("rand", "" +
random_mesimeriano);
        random_bradyno = new
Random().nextInt(bradyna.size());

```

```

        Log.i("rand", "" + random_bradyνο);

        textView_25.setText(""
            +
prwina.get(random_prwino).getNamePrwina());
        textView_27.setText(""
            +
mesimeriana.get(random_mesimeriano)
            .getNameMesimeriana());
        textView_29.setText(""
            +
bradyνο.get(random_bradyνο).getNameBradyνο());
        sum_calories =
prwina.get(random_prwino)
            .getThermidesPrwino()
            +
mesimeriana.get(random_mesimeriano)
            .getThermidesMesimeriana()
            +
bradyνο.get(random_bradyνο)
            .getThermidesBradyνο());
        textView_31.setText("" + sum_calories);
    }
    if (host_2.getCurrentTab() == 6) {
        random_prwino = new
Random().nextInt(prwina.size());
        Log.i("rand", "" + random_prwino);
        random_mesimeriano = new
Random().nextInt(mesimeriana
            .size());

```

```

random_mesimeriano);
Random().nextInt(bradyna.size());
Log.i("rand", "" + random_bradyo);
textV_25.setText(""
+
prwina.get(random_prwino).getNamePrwina());
textV_27.setText(""
+
mesimeriana.get(random_mesimeriano)
.getNameMesimeriana());
textV_29.setText(""
+
bradyna.get(random_bradyo).getNameBradyna());
sum_calories =
prwina.get(random_prwino)
.getThermidesPrwino()
+
mesimeriana.get(random_mesimeriano)
.getThermidesMesimeriana()
+
bradyna.get(random_bradyo)
.getThermidesBradyna());
textV_31.setText("" + sum_calories);
}
}
});

```

```

        tableView_1 = host_2.getTabWidget().getChildAt(0);
        tableView_2 = host_2.getTabWidget().getChildAt(1);
        tableView_3 = host_2.getTabWidget().getChildAt(2);
        tableView_4 = host_2.getTabWidget().getChildAt(3);
        tableView_5 = host_2.getTabWidget().getChildAt(4);
        tableView_6 = host_2.getTabWidget().getChildAt(5);
        tableView_7 = host_2.getTabWidget().getChildAt(6);

        tableView_1.setOnLongClickListener(new
OnLongClickListener() {

            @Override
            public boolean onLongClick(View v) {
                // TODO Auto-generated method stub
                if (host_2.getCurrentTab() == 0) {
                    Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                    // intent gia insert event

                    calIntent.setType("vnd.android.cursor.item/event");

                    calIntent.putExtra(Events.TITLE,
"Monday-Eating");

                    calIntent.putExtra(
                        Events.DESCRPTION,
                        "Breakfast : " +
textView_25.getText()
                        + ".. Lunch
: " + textView_27.getText()
                        + "..
Dinner : " + textView_29.getText()
                        + ".. Total
Calories : "

```

```
textV_31.getText());  
  
calendar = Calendar.getInstance();  
Log.i("date",  
calendar.get(calendar.YEAR) + " "  
+  
(calendar.get(calendar.MONTH) + 1) + " "  
+  
calendar.get(calendar.DAY_OF_MONTH));  
// ta years einai atousia , oi mines  
metrane apo 0-11  
  
// kai oi  
// days autousies  
while  
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.MONDAY) {  
    calendar.add(calendar.DATE, 1);  
}  
// default timi gia to pote tha  
arxisei na metraei  
  
calIntent.putExtra(  
    CalendarContract.EXTRA_EVENT_ALL_DAY, true);  
calIntent.putExtra(  
    CalendarContract.EXTRA_EVENT_BEGIN_TIME,  
    calendar.getTimeInMillis());  
calIntent.putExtra(  
    CalendarContract.EXTRA_EVENT_END_TIME,  
    calendar.getTimeInMillis());  
startActivity(calIntent);
```

```

        }
        return false;
    }

});

    tableView_2.setOnLongClickListener(new
OnLongClickListener() {

        @Override
        public boolean onLongClick(View v) {
            // TODO Auto-generated method stub
            if (host_2.getCurrentTab() == 1) {
                Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                // intent gia insert event

                calIntent.setType("vnd.android.cursor.item/event");
                calIntent.putExtra(Events.TITLE,
"Tuesday-Eating");

                calIntent.putExtra(
                    Events.DESCRPTION,
                    "Breakfast : " +
                    textView_25.getText()
                    + ".. Lunch
                    : " + textView_27.getText()
                    + "..
                    Dinner : " + textView_29.getText()
                    + ".. Total
                    Calories : "
                    +
                    textView_31.getText());

                calendar = Calendar.getInstance();

```



```
calendar.get(calendar.YEAR) + " "      Log.i("date",
                                       +
(calendar.get(calendar.MONTH) + 1) + " "
                                       +
calendar.get(calendar.DAY_OF_MONTH));
// ta years einai atousia , oi mines
metrane apo 0-11
// kai oi
// days autousies
while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.TUESDAY)
    calendar.add(calendar.DATE, 1);
// default timi gia to pote tha
arxisei na metraei
    calIntent.putExtra(
        CalendarContract.EXTRA_EVENT_ALL_DAY, true);
    calIntent.putExtra(
        CalendarContract.EXTRA_EVENT_BEGIN_TIME,
        calendar.getTimeInMillis());
    calIntent.putExtra(
        CalendarContract.EXTRA_EVENT_END_TIME,
        calendar.getTimeInMillis());
        startActivity(calIntent);
    }
return false;
}
```

```

    });
    tabView_3.setOnLongClickListener(new
OnLongClickListener() {

        @Override
        public boolean onLongClick(View v) {
            // TODO Auto-generated method stub
            if (host_2.getCurrentTab() == 2) {
                Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                // intent gia insert event

                calIntent.setType("vnd.android.cursor.item/event");
                calIntent.putExtra(Events.TITLE,
"Wednesday-Eating");

                calIntent.putExtra(
                    Events.DESCRPTION,
                    "Breakfast : " +
                    textV_25.getText()
                    + ".. Lunch
                    : " + textV_27.getText()
                    + "..
                    Dinner : " + textV_29.getText()
                    + ".. Total
                    Calories : "
                    +
                    textV_31.getText());

                calendar = Calendar.getInstance();
                Log.i("date",
                    calendar.get(calendar.YEAR) + " "
                    +
                    (calendar.get(calendar.MONTH) + 1) + " "

```

```
calendar.get(calendar.DAY_OF_MONTH));
// ta years einai atousia , oi mines
metrane apo 0-11
// kai oi
// days autousies
while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.WEDNESDAY)
calendar.add(calendar.DATE, 1);
// default timi gia to pote tha
arxisei na metraei
calIntent.putExtra(
CalendarContract.EXTRA_EVENT_ALL_DAY, true);
calIntent.putExtra(
CalendarContract.EXTRA_EVENT_BEGIN_TIME,
calendar.getTimeInMillis());
calIntent.putExtra(
CalendarContract.EXTRA_EVENT_END_TIME,
calendar.getTimeInMillis());
startActivity(calIntent);
}
return false;
}
});
```

```

        tabView_4.setOnLongClickListener(new
OnLongClickListener() {

        @Override

        public boolean onLongClick(View v) {
            // TODO Auto-generated method stub
            if (host_2.getCurrentTab() == 3) {
                Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                // intent gia insert event

                calIntent.setType("vnd.android.cursor.item/event");

                calIntent.putExtra(Events.TITLE,
"Thursday-Eating");

                calIntent.putExtra(
                    Events.DESCRPTION,
                    "Breakfast : " +
textV_25.getText()
                    + ".. Lunch
: " + textV_27.getText()
                    + "..
Dinner : " + textV_29.getText()
                    + ".. Total
Calories : "
                    +
textV_31.getText());

                calendar = Calendar.getInstance();
                Log.i("date",
calendar.get(calendar.YEAR) + " "
                    +
(calendar.get(calendar.MONTH) + 1) + " "
                    +
calendar.get(calendar.DAY_OF_MONTH));

                // ta years einai atousia , oi mines
metrane apo 0-11

```

```
        // kai oi
        // days autousies
        while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.THURSDAY)
            calendar.add(calendar.DATE, 1);
        // default timi gia to pote tha
        arxisei na metraei
            calIntent.putExtra(
                CalendarContract.EXTRA_EVENT_ALL_DAY, true);
            calIntent.putExtra(
                CalendarContract.EXTRA_EVENT_BEGIN_TIME,
                calendar.getTimeInMillis());
            calIntent.putExtra(
                CalendarContract.EXTRA_EVENT_END_TIME,
                calendar.getTimeInMillis());
            startActivity(calIntent);
        }
        return false;
    }
});
tabView_5.setOnLongClickListener(new
OnLongClickListener() {
    @Override
```

```

        public boolean onLongClick(View v) {
            // TODO Auto-generated method stub
            if (host_2.getCurrentTab() == 4) {
                Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                // intent gia insert event

                calIntent.setType("vnd.android.cursor.item/event");
                calIntent.putExtra(Events.TITLE,
"Friday-Eating");

                calIntent.putExtra(
                    Events.DESCRPTION,
                    "Breakfast : " +
textV_25.getText()
                    + ".. Lunch
: " + textV_27.getText()
                    + "..
Dinner : " + textV_29.getText()
                    + ".. Total
Calories : "
                    +
textV_31.getText());

                calendar = Calendar.getInstance();
                Log.i("date",
calendar.get(calendar.YEAR) + " "
                    +
(calendar.get(calendar.MONTH) + 1) + " "
                    +
calendar.get(calendar.DAY_OF_MONTH));

                // ta years einai atousia , oi mines
metrane apo 0-11

                // kai oi
                // days autousies

                while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.FRIDAY)

```

```
        calendar.add(calendar.DATE, 1);
        // default timi gia to pote tha
arxisei na metraei
        calIntent.putExtra(
            CalendarContract.EXTRA_EVENT_ALL_DAY, true);
        calIntent.putExtra(
            CalendarContract.EXTRA_EVENT_BEGIN_TIME,
            calendar.getTimeInMillis());
        calIntent.putExtra(
            CalendarContract.EXTRA_EVENT_END_TIME,
            calendar.getTimeInMillis());

        startActivity(calIntent);
    }

    return false;
}

});
tabView_6.setOnLongClickListener(new
OnLongClickListener() {

    @Override
    public boolean onLongClick(View v) {
        // TODO Auto-generated method stub
        if (host_2.getCurrentTab() == 5) {
```

```

Intent(Intent.ACTION_INSERT);
                                Intent calIntent = new
                                // intent gia insert event

                                calIntent.setType("vnd.android.cursor.item/event");
                                calIntent.putExtra(Events.TITLE,
"Saturday-Eating");
                                calIntent.putExtra(
                                Events.DESCRPTION,
                                "Breakfast : " +
textV_25.getText()
                                + ".. Lunch
: " + textV_27.getText()
                                + "..
Dinner : " + textV_29.getText()
                                + ".. Total
Calories : "
                                +
textV_31.getText());
                                calendar = Calendar.getInstance();
                                Log.i("date",
calendar.get(calendar.YEAR) + " "
                                +
(calendar.get(calendar.MONTH) + 1) + " "
                                +
calendar.get(calendar.DAY_OF_MONTH));
                                // ta years einai atousia , oi mines
metrane apo 0-11
                                // kai oi
                                // days autousies
                                while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.SATURDAY)
                                calendar.add(calendar.DATE, 1);
                                // default timi gia to pote tha
arxisei na metraei

```



```
                calIntent.putExtra(
                    CalendarContract.EXTRA_EVENT_ALL_DAY, true);
                calIntent.putExtra(
                    CalendarContract.EXTRA_EVENT_BEGIN_TIME,
                    calendar.getTimeInMillis());
                calIntent.putExtra(
                    CalendarContract.EXTRA_EVENT_END_TIME,
                    calendar.getTimeInMillis());

                startActivity(calIntent);
            }

            return false;
        }

    });
    tableView_7.setOnLongClickListener(new
OnLongClickListener() {

        @Override
        public boolean onLongClick(View v) {
            // TODO Auto-generated method stub
            if (host_2.getCurrentTab() == 6) {
                Intent calIntent = new
Intent(Intent.ACTION_INSERT);

                // intent gia insert event
```

```

        calIntent.setType("vnd.android.cursor.item/event");
        calIntent.putExtra(Events.TITLE,
"Sunday-Eating");
        calIntent.putExtra(
            Events.DESCRPTION,
            "Breakfast : " +
textV_25.getText()
            + ".. Lunch
: " + textV_27.getText()
            + "..
Dinner : " + textV_29.getText()
            + ".. Total
Calories : "
            +
textV_31.getText());
        calendar = Calendar.getInstance();
        Log.i("date",
calendar.get(calendar.YEAR) + " "
            +
(calendar.get(calendar.MONTH) + 1) + " "
            +
calendar.get(calendar.DAY_OF_MONTH));
        // ta years einai atousia , oi mines
        // kai oi
        // days autousies
        while
(calendar.get(calendar.DAY_OF_WEEK) != Calendar.SUNDAY)
            calendar.add(calendar.DATE, 1);
        // default timi gia to pote tha
        arxisei na metraei
        calIntent.putExtra(
            CalendarContract.EXTRA_EVENT_ALL_DAY, true);

```

```
        calIntent.putExtra(  
  
        CalendarContract.EXTRA_EVENT_BEGIN_TIME,  
  
        calendar.getTimeInMillis());  
  
        calIntent.putExtra(  
  
        CalendarContract.EXTRA_EVENT_END_TIME,  
  
        calendar.getTimeInMillis());  
  
        startActivity(calIntent);  
    }  
  
    return false;  
}  
  
});  
}  
  
}
```