

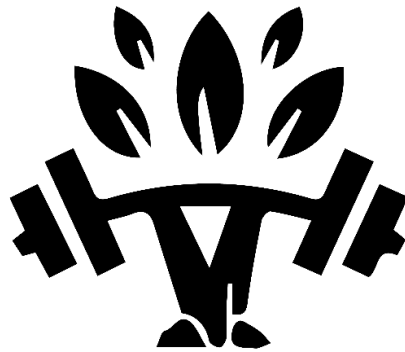


ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Εφαρμογή σωματικής άσκησης για κινητά τηλέφωνα
Android»



Της φοιτήτριας
Έλλης Στεργιάδου
Αρ. Μητρώου: 144228

Επιβλέπων
Ονοματεπώνυμο Ευκλείδης
Κεραμόπουλος
Αναπληρωτής Καθηγητής

Ημερομηνία 6/7/2021

Τίτλος Π.Ε. **Εφαρμογή υγείας για κινητά τηλέφωνα Android**

Κωδικός Π.Ε. **20240**

Όνοματεπώνυμο φοιτήτριας **Έλλη Στεργιάδου**

Όνοματεπώνυμο εισηγητή **Ευκλείδης Κεραμόπουλος**

Ημερομηνία ανάληψης Π.Ε. **10-11-2020**

Ημερομηνία περάτωσης Π.Ε. **6-7-2021**

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Έλλης Στεργιάδου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Την περίοδο αναζήτησης του θέματος της πτυχιακής εργασίας, συνέβη ο κορωνοϊός. Ως ένα δραστήριο άτομο που αγαπά την άσκηση, αναζητούσα τρόπους για να αθληθώ με ασφάλεια. Ο προπονητής όμως είναι ένα σημαντικό κομμάτι της άσκησης καθώς καθοδηγεί τον αθλητή και του δίνει το πρόγραμμα που χρειάζεται. Επέλεξα λοιπόν να αναπτύξω μια εφαρμογή η οποία θα παρακολουθεί την καθημερινή μου σωματική άσκηση και θα μου δίνει κίνητρο για «να σηκωθώ από τον καναπέ». Επίσης θα με καθοδηγεί με ασκήσεις και στόχους.

Κατά τη διάρκεια της αναζήτησης, επιλέχθηκα να είμαι προπονήτρια μιας παιδικής ομάδας Ρομποτικής και Καινοτομίας, η οποία θα διαγωνιζόταν σε αντίστοιχο Πανελλήνιο και Παγκόσμιο Διαγωνισμό. Το φετινό εγχείρημα ήταν να παρουσιάσουμε ένα πρότζεκτ το οποίο θα δίνει κίνητρο στους πολίτες να κινηθούν. Έτσι σκεφτήκαμε να δημιουργήσουμε ένα οικολογικό-πράσινο πάρκο εκγύμνασης με υπόγειες θέσεις στάθμευσης. Η εφαρμογή που ανέπτυξα θα μετράει την δραστηριότητα των πολιτών και θα επιβραβεύει τους πολίτες με δωρεάν χρόνο στάθμευσης.

Η εφαρμογή βοηθάει στο να φροντίζουν οι χρήστες τον εαυτό τους και να πετυχαίνουν στόχους που αφορούν τη σωματική κατάστασή τους.

Περίληψη

Στην δύσκολη εποχή που διανύουμε είναι αδύνατη η εκγύμναση σε γυμναστήρια και σε άλλους αθλητικούς χώρους. Η σωματική άσκηση όμως είναι σημαντική για την υγεία του ανθρώπου, τόσο για το σώμα όσο και για το μυαλό. Σκοπός της πτυχιακής εργασίας είναι η δημιουργία μιας Android εφαρμογής η οποία καταγράφει τη σωματική άσκηση αλλά και δίνει κίνητρο στον χρήστη να αθληθεί, με την χρήση της Android συσκευής του και των ειδικών αισθητήρων που αυτή περιέχει. Στην εφαρμογή θα γίνει ενδεικτικά χρήση των αισθητήρων κίνησης και θέσης. Επίσης, ενδεικτικά κάποιες από τις υπηρεσίες που θα παρέχονται είναι: (α) Καταμέτρηση βημάτων, (β) Καταμέτρηση χιλιομέτρων που διανύθηκαν, (γ) Καταμέτρηση θερμίδων, (δ) Προβολή χάρτη και διαδρομών, (ε) Προβολή ασκήσεων εκγύμνασης, (στ) Καθημερινοί ατομικοί στόχοι.

«Android Fitness App»

«Elli Stergiadou»

Abstract

During the Covid-19 period and quarantines, it is impossible to work out in gyms and other sports venues. But physical exercise is important for health, both for the body and for the mind. This thesis aims to develop an Android application that tracks physical exercise and motivates the user to exercise, using his Android device and the special sensors it contains. The application uses Internet to save and update users' data, and also the accelerometer and position sensors. Some of the services it provides are:

- Tracking exercise
- Monitoring health
- Counting steps
- Counting exercise kilometers
- Counting burned calories
- View map and routes
- View fitness exercises
- Set Personal goals

Ευχαριστίες

Ευχαριστώ την οικογένεια μου για την υποστήριξη όλα αυτά τα χρόνια των σπουδών μου, τους φίλους αθλητές και μη, και “last but not least” τον καθηγητή μου Ευκλείδη Κεραμόπουλο ο οποίος με τη διδασκαλία των «Αλληλεπίδραση Ανθρώπου Μηχανής» και «Προηγμένη Αλληλεπίδραση Ανθρώπου Μηχανής- Προγραμματισμός Κινητών Συσκευών», μου έδωσε τα πρώτα βήματα καθώς και το κίνητρο για να ασχοληθώ με την Ανάπτυξη λογισμικού Android και το UX/UI.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη	vi
Abstract.....	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xii
Κατάλογος Εικόνων.....	xii
Κατάλογος Πινάκων	xiv
Συντομογραφίες.....	xv
Κεφάλαιο 1ο: Εισαγωγή στην Εφαρμογή.....	1
1.1 Εισαγωγή	1
1.2 Δομή Πτυχιακής Εργασίας.....	1
1.3 Έρευνα.....	1
1.3.1 Ανάλυση Αποτελεσμάτων.....	2
1.4 Μελέτη αποτελεσμάτων έρευνας	7
1.5 Στόχοι εφαρμογής.....	8
1.6 Λειτουργίες Εφαρμογής.....	8
1.6.1 Δημιουργία Προφίλ	9
1.6.2 Καταμέτρηση βημάτων.....	9
1.6.3 Καταμέτρηση χιλιομέτρων που διανύθηκαν	9
1.6.4 Καταμέτρηση καμένων θερμίδων.....	9
1.6.5 Καταμέτρηση κατανάλωσης νερού.....	9
1.6.6 Καταμέτρηση σωματικού βάρους.....	9
1.6.7 Εύρεση BMI.....	10
1.6.8 Θέση και επίτευξη στόχων	10
1.6.9 Προβολή χάρτη και εντοπισμός τοποθεσίας	10
1.6.10 Προβολή ασκήσεων.....	10
1.7 Επίλογος.....	11
Κεφάλαιο 2ο: Τεχνολογίες που χρησιμοποιήθηκαν.....	13
2.1 Εισαγωγή	13
2.1 Android Studio	13
2.2 Λειτουργικό Android.....	13

2.3	Αρχιτεκτονική Android	13
2.3.1	Activity Manager	15
2.4	Activities και Fragments	16
2.5	Αισθητήρες	19
2.5.1	Αισθητήρες κίνησης.....	19
2.5.2	Αισθητήρας Θέσης.....	19
2.6	Google Maps JavaScript API.....	19
2.7	Facebook SDK	20
2.8	Firebase Firestore	21
2.9	UX/UI	22
2.9.1	Διαδικασία σχεδίασης UX.....	22
2.9.2	Σχεδιασμός Διεπαφών.....	24
Κεφάλαιο 3ο:	Η εφαρμογή	27
3.1	Εισαγωγή	27
3.2	Κύκλος ζωής ανάπτυξης εφαρμογών για κινητά (MADLC)	29
3.3	Φάση σχεδίασης	29
3.4	Συλλογές Βάσης Δεδομένων.....	30
3.4.1	Users	30
3.4.2	Metrics	31
3.4.3	Water.....	31
3.4.4	Records.....	32
3.4.5	Goals	32
3.5	Δομή πρότζεκτ στο Android Studio	32
3.6	AndroidManifest.xml.....	33
3.6.1	Δικαιώματα.....	35
3.7	Βιβλιοθήκες.....	36
3.7.1	GOOGLE PLAY SERVICES	36
3.7.2	AndroidX test	36
3.7.3	Facebook Login for Android	36
3.7.4	Material Design	36
3.7.5	Animation.....	38
	Circle images	40
3.8	Gradle	40
3.8.1	Γενικά.....	40
3.9	Activities	41

3.9.1	Launch Activity	41
3.9.2	Sign Up Activity	43
3.9.3	BMI Calculator Activity	45
3.9.4	Home Activity	46
3.9.5	Water Activity	47
3.9.6	Contact Us	49
3.10	Fragments.....	49
3.10.1	Home Fragment	49
3.10.2	Goals Fragment.....	52
3.10.3	Out Training Fragment.....	54
3.10.4	Stretch Fragment.....	58
3.11	App resources.....	58
3.11.1	Menu	58
3.11.2	Drawer shift- Side Menu	59
3.11.3	Progress Bar	61
Κεφάλαιο 4ο:	Συμπεράσματα ή/και προτάσεις βελτίωσης.....	65
BIBΛΙΟΓΡΑΦΙΑ		66

Κατάλογος Σχημάτων

Σχήμα 1.1 Συχνότητα άσκησης ανά βδομάδα	2
Σχήμα 1.2 Χρήση κινητού για καταγραφή σωματικής άσκησης.....	3
Σχήμα 1.3 Χρήση κινητού για καθοδήγηση σωματικής άσκησης.....	3
Σχήμα 1.4 Κατανάλωση νερού σε λίτρα	4
Σχήμα 1.5 Τοποθεσία άσκησης προ Covid-19	4
Σχήμα 1.6 Τοποθεσία άσκησης κατά την περίοδο καραντίνας Covid-19	5
Σχήμα 1.7 Τοποθεσία άσκησης μετά την περίοδο καραντίνας Covid-19	5
Σχήμα 1.8 Σταμάτησαν τη γυμναστική εν μέσω καραντίνας	5
Σχήμα 1.9 Ασκούνταν πιο εντατικά στην καραντίνα	6
Σχήμα 1.10 Συνέβαλε ο Covid-19 στον να ασκείστε πιο πολύ	6
Σχήμα 1.11 Άσκηση για βελτίωση υγείας.....	7
Σχήμα 1.12 Άσκηση για βελτίωση εμφάνισης	7
Σχήμα 1.13 Σχέση χώρου άσκησης πριν την καραντίνα με την παύση άσκησης.....	8
Σχήμα 2.1 Αρχιτεκτονική Android Συστήματος	14
Σχήμα 2.2 Activity Lifecycle [27]	16
Σχήμα 2.3 Κύκλος ζωής Fragment	17
Σχήμα 3.1 Mobile Application Development Lifecycle (MADLC) [19]	29
Σχήμα 3.2 Figma wireframe #1	30
Σχήμα 3.3 Δομή Android πρότζεκτ	33
Σχήμα 3.4 Material Components [20].....	37
Σχήμα 3.5 Navigation drawers και tabs [20].....	38
Σχήμα 3.6 Gradle Scripts	40
Σχήμα 3.7 Dependencies	41
Σχήμα 3.8 Άξονες συντεταγμένων XYZ [26]	50

Κατάλογος Εικόνων

Εικόνα 1.1 Πρόταση Καταχώρησης Βάρους και Ύψος.....	10
Εικόνα 1.2 Διεπαφή Υπολογισμού BMI	10
Εικόνα 2.1 Firebase methods.....	22
Εικόνα 2.2 Behance	23
Εικόνα 2.3 Lo-fi Storyboard [39]	24
Εικόνα 2.4 Figma.....	25
Εικόνα 3.1 Διεπαφές πρώτης έκδοσης της εφαρμογής	27
Εικόνα 3.2 Διεπαφή "Home"	28
Εικόνα 3.3 "Side Menu".....	28
Εικόνα 3.4 Διεπαφή "Goals"	28
Εικόνα 3.5 Διεπαφή "Training"	28
Εικόνα 3.6 Διεπαφή "Run"	28
Εικόνα 3.7 Διεπαφή "Stretch"	28
Εικόνα 3.8 Collection "Users".....	31
Εικόνα 3.9 Collection "Metrics"	31

Εικόνα 3.10 Collection “Water”	31
Εικόνα 3.11 Collection “Records”	32
Εικόνα 3.12 Collection “Goals”	32
Εικόνα 3.13 Δομή manifest τμήμα 1	34
Εικόνα 3.14 Δομή manifest τμήμα 2	34
Εικόνα 3.15 meta-data	35
Εικόνα 3.16 Permissions	35
Εικόνα 3.17 XML κώδικας με το Lottie Animation	38
Εικόνα 3.18 Μέθοδοι GIFLib [22]	39
Εικόνα 3.19 Java μέθοδος playAnimation()	39
Εικόνα 3.20 Ενεργοποίηση playAnimation() (γραφικά)	40
Εικόνα 3.21 Main Launcher	41
Εικόνα 3.22 Διεπαφή Launch Activity	42
Εικόνα 3.23 Αρχικοποίηση Facebook & Google buttons	42
Εικόνα 3.24 Μέθοδοι σύνδεσης μέσω Facebook & Google	43
Εικόνα 3.25 Διεπαφή Sign Up Activity	44
Εικόνα 3.26 Birthday Calendar	44
Εικόνα 3.27 Calendar Αρχικοποίηση και DatePicker	44
Εικόνα 3.28 Ενημέρωση Βάσης και εμφάνιση BMI αποτελέσματος	45
Εικόνα 3.29 Υπολογισμός BMI	45
Εικόνα 3.30 Χτίσιμο bottom_menu	46
Εικόνα 3.31 Μέθοδος επιλογής αντικειμένου στο Bottom Menu	46
Εικόνα 3.32 Μέθοδος navigationDrawer()	47
Εικόνα 3.33 Μέθοδοι item selected and logout	47
Εικόνα 3.34 Εισαγωγή δεδομένων στο Collection Water	48
Εικόνα 3.35 Ενημέρωση δεδομένων εφαρμογής από την βάση δεδομένων	48
Εικόνα 3.36 Μέθοδος αποστολής ηλεκτρονικών μηνυμάτων	49
Εικόνα 3.37 Μέθοδος onSensorChanged()	50
Εικόνα 3.38 Μαθηματική συνάρτηση για την εύρεση του Magnitude	51
Εικόνα 3.39 Σύνδεση με τη βάση	51
Εικόνα 3.40 Εισαγωγή δεδομένων στη βάση	51
Εικόνα 3.41 Lifecycle μέθοδοι onResume(), onStop()	52
Εικόνα 3.42 Διεπαφή Goals	52
Εικόνα 3.43 Water & Calories Spinners XML	53
Εικόνα 3.44 Spinner Java	53
Εικόνα 3.45 Γραφικό κομμάτι Spinner	54
Εικόνα 3.46 Map Frame XML	54
Εικόνα 3.47 Έλεγχος δικαιώματος τοποθεσίας	55
Εικόνα 3.48 Μέθοδος onMapReady()	55
Εικόνα 3.49 Άρνηση παραχώρησης δικαιώματος τοποθεσίας	56
Εικόνα 3.50 Παραχώρηση δικαιώματος τοποθεσίας	56
Εικόνα 3.51 Μέθοδος αλλαγής κουμπιού Start-Stop	56
Εικόνα 3.52 Μέθοδοι Start και Format Timer	57
Εικόνα 3.53 Reset timer	57
Εικόνα 3.54 Stretch Fragment	58
Εικόνα 3.55 XML Bottom Menu items	59

Εικόνα 3.56 Γραφικό κομμάτι Bottom Menu.....	59
Εικόνα 3.57 XML Drawer Shift Menu	60
Εικόνα 3.58 Γραφικό κομμάτι Drawer Shift Menu	60
Εικόνα 3.59 Εμφάνιση Drawer Shift	60
Εικόνα 3.60 Μέθοδος onNavigationItemSelectedListener()	61
Εικόνα 3.61 Μέθοδος setProgress().....	61
Εικόνα 3.62 Stepgoal=6000	62
Εικόνα 3.63 Stepgoal=1000	62
Εικόνα 3.64 Values resources	62
Εικόνα 3.65 Colors	63

Κατάλογος Πινάκων

Πίνακας 2.1 Διαφορές χαρακτηριστικών Activity-Fragment [29]	18
Πίνακας 2.2 Διαφορές Intent Activity-Fragment	18
Πίνακας 2.3 Διαφορές Casting Activity-Fragment.....	18
Πίνακας 2.4 Γενική ψυχολογία χρώματος	23

Συντομογραφίες

UX	User Experience
UI	User Interface
Lo-fi	Low fidelity
Hi-fi	High Fidelity
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
XML	Extensive Markup Language
API	Application Programming Interface
SDK	Software Development Kit

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Κατά τη διάρκεια της περιόδου Covid-19 και της καραντίνας, η άσκηση σε γυμναστήρια και άλλους αθλητικούς χώρους είναι αδύνατη να γίνει. Όμως η σωματική άσκηση είναι σημαντική για την υγεία, τόσο για το σώμα, όσο για το μυαλό και την ψυχολογία. Ο σκοπός της πτυχιακής εργασίας μου ήταν η ανάπτυξη μιας εφαρμογής Android που παρακολουθεί τη σωματική άσκηση και παρακινεί τον χρήστη να ασκηθεί, χρησιμοποιώντας τη συσκευή του Android και τους ειδικούς αισθητήρες που περιέχει. Η εφαρμογή χρησιμοποιεί το Διαδίκτυο για να αποθηκεύει τα δεδομένα των χρηστών και να ενημερώνει τη βάση δεδομένων, καθώς και το επιταχυνσιόμετρο και τον αισθητήρα θέσης. Μερικές από τις υπηρεσίες που παρέχει είναι:

- Παρακολούθηση σωματικής άσκησης
- Παρακολούθηση της υγείας
- Καταμέτρηση βημάτων
- Καταμέτρηση χιλιομέτρων άσκησης
- Καταμέτρηση καμένων θερμίδων
- Προβολή χάρτη και διαδρομών
- Προβολή ασκήσεων γυμναστικής
- Ορισμός και κατάκτηση προσωπικών στόχων

1.2 Δομή Πτυχιακής Εργασίας

Στο πρώτο κεφάλαιο της πτυχιακής εργασίας γίνεται μια γενική αναφορά στην εφαρμογή, περιγράφονται οι στόχοι της και παρουσιάζονται τα αποτελέσματα της έρευνας που εκπονήθηκε για την κατανόηση των αναγκών των χρηστών για μια εφαρμογή καταγραφής και καθοδήγησης σωματικής άσκησης. Στο δεύτερο κεφάλαιο αναλύονται οι τεχνολογίες και οι πλατφόρμες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Το τρίτο κεφάλαιο εμβαθύνει στην κύρια πλατφόρμα Android, στην οποία αναπτύχθηκε ο κώδικας της εφαρμογής και γίνεται μια αναφορά των συστατικών της. Τέλος στο τέταρτο κεφάλαιο αναφέρονται τα συμπεράσματα.

1.3 Έρευνα

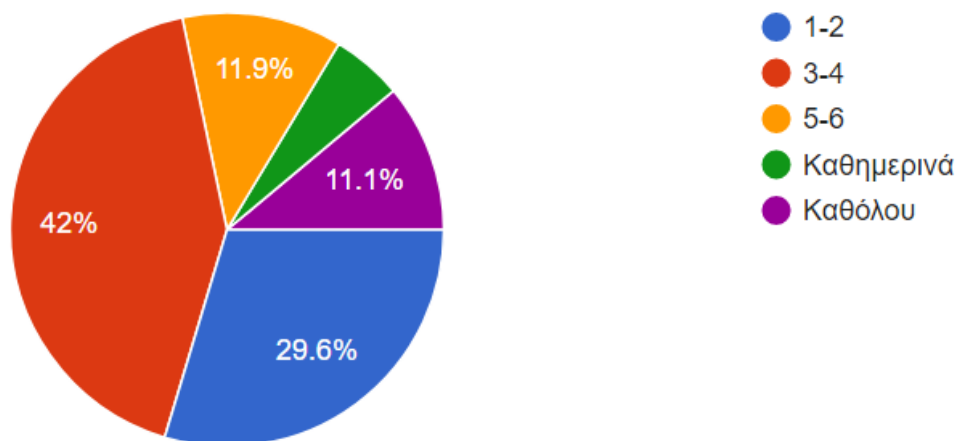
Στόχος μου ήταν η ανάπτυξη μιας δημοφιλούς, χρήσιμης, αποτελεσματικής και σύγχρονης εφαρμογής υγείας. Γι' αυτό το λόγο, κατά τη φάση της ταυτοποίησης και της συλλογής ιδεών, σκέφτηκα πως η εκπόνηση μιας έρευνας για την σωματική άσκηση και την χρήση κινητών συσκευών για την καταγραφή της και για τις συνήθειες σχετικά με την σωματική άσκηση των ατόμων κατά τη εποχή του Covid-19, θα μου ήταν αρκετά χρήσιμη. Στην έρευνα έλαβαν μέρος 240 άτομα, 128 άντρες και 112 γυναίκες, ηλικίας 18-55 ετών.

1.3.1 Ανάλυση Αποτελεσμάτων

Αρχικά οι συμμετέχοντες στην έρευνα απάντησαν σε βασικές προσωπικές ερωτήσεις όπως το φύλο και η ηλικία, για να μελετηθεί κατά πόσο παίζουν ρόλο αυτές οι πληροφορίες στη σωματική άσκηση. Έπειτα οι βασικές ερωτήσεις ήταν οι παρακάτω.

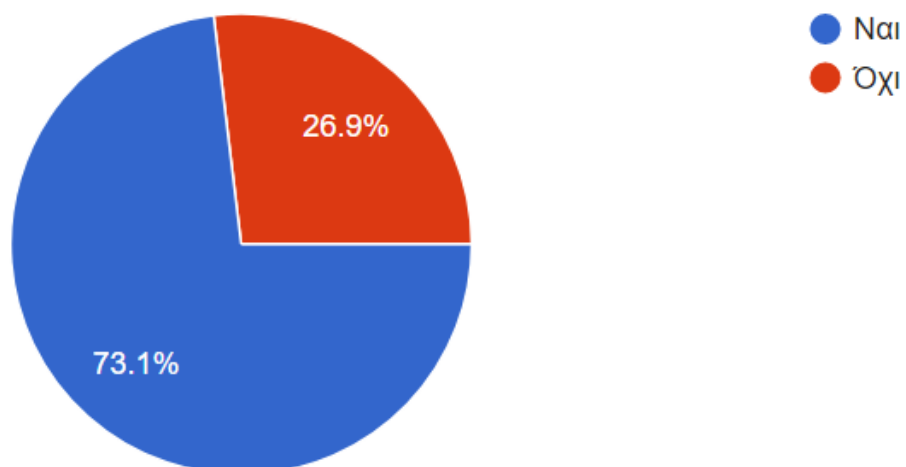
1. Πόσες φορές τη βδομάδα ασκείστε;
2. Πόσες ώρες ασκείστε ημερησίως;
3. Είδος άσκησης
4. Χρησιμοποιείτε/ θα χρησιμοποιούσατε κάποια εφαρμογή στο κινητό σας για την καταγραφή σωματικής άσκησης;
5. Χρησιμοποιείτε/ θα χρησιμοποιούσατε κάποια εφαρμογή στο κινητό σας για οδηγό σωματικής άσκησης;
6. Πόσα λίτρα νερό καταναλώνετε ημερησίως; (λίτρα)
7. Που προτιμούσατε να ασκείστε πριν τον Covid-19;
8. Κατά τη διάρκεια της καραντίνας, που επιλέγατε να ασκείστε;
9. Μετά τη λήξη της καραντίνας, που προτιμάτε να ασκείστε;
10. Κατά τη διάρκεια της καραντίνας, σταματήσατε να ασκείστε;
11. Κατά τη διάρκεια της καραντίνας, ασκούσασταν πιο εντατικά;
12. Ασκείστε για να βελτιώσετε την υγεία σας;
13. Ασκείστε για να βελτιώσετε την εμφάνισή σας;
14. Θεωρείτε ότι ο Covid-19 συνέβαλε στο να προσέχετε την υγεία σας και να ασκείστε περισσότερο;

Όπως παρατηρούμε στο Σχήμα 1.1, το 41.5% των ερωτηθέντων, ασκείτε ανεπαρκώς ή δεν ασκείται καθόλου.

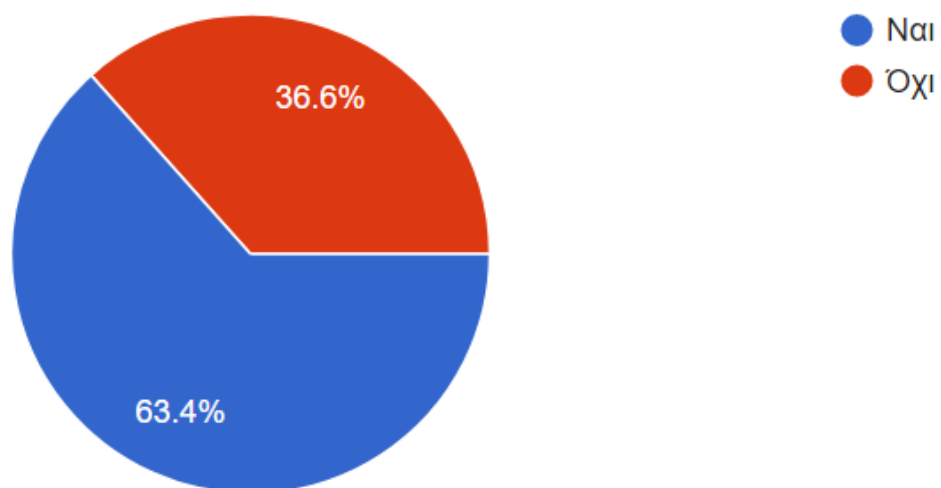


Σχήμα 1.1 Συχνότητα άσκησης ανά εβδομάδα

Το μεγαλύτερο ποσοστό των ατόμων επιθυμούσε να χρησιμοποιήσει μια εφαρμογή στο κινητό του για καταγραφή (Σχήμα 1.2) και οδηγό σωματικής άσκησης (Σχήμα 1.3)

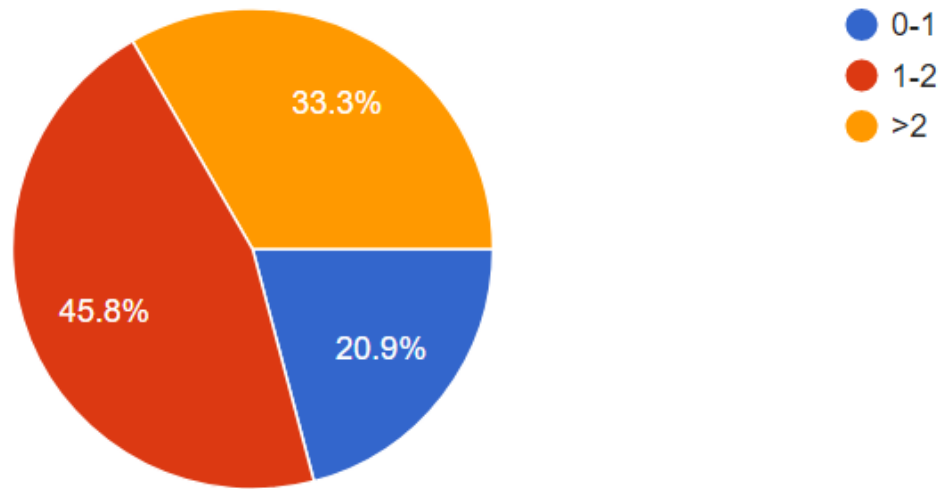


Σχήμα 1.2 Χρήση κινητού για καταγραφή σωματικής άσκησης



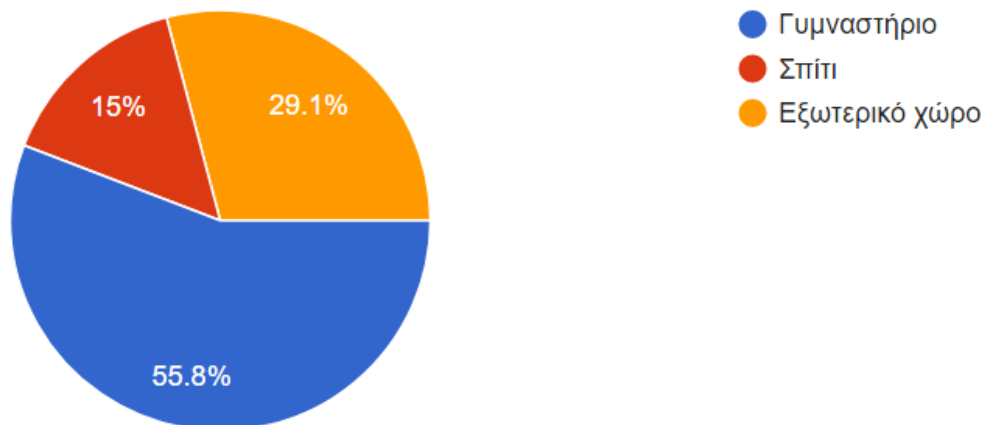
Σχήμα 1.3 Χρήση κινητού για καθοδήγηση σωματικής άσκησης

Όσον αφορά την κατανάλωση νερού που είναι σημαντικό για το σώμα μας, είδα ότι μόνο το 1/3 πίνει αρκετό νερό ημερησίως (Σχήμα 1.4).

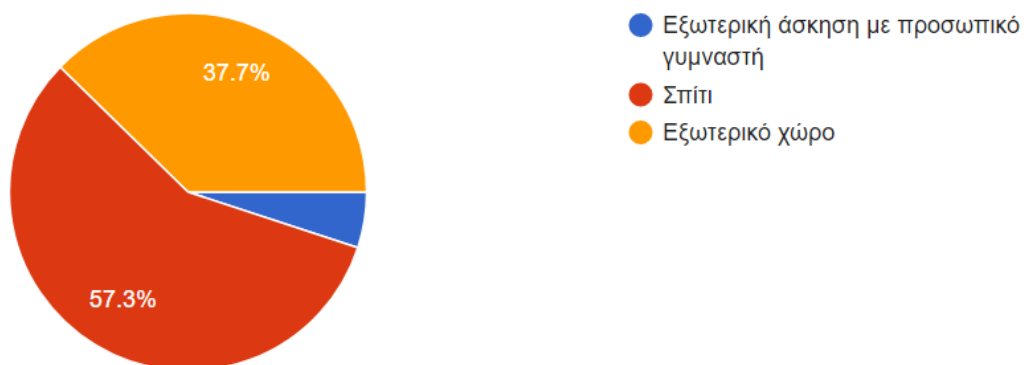


Σχήμα 1.4 Κατανάλωση νερού σε λίτρα

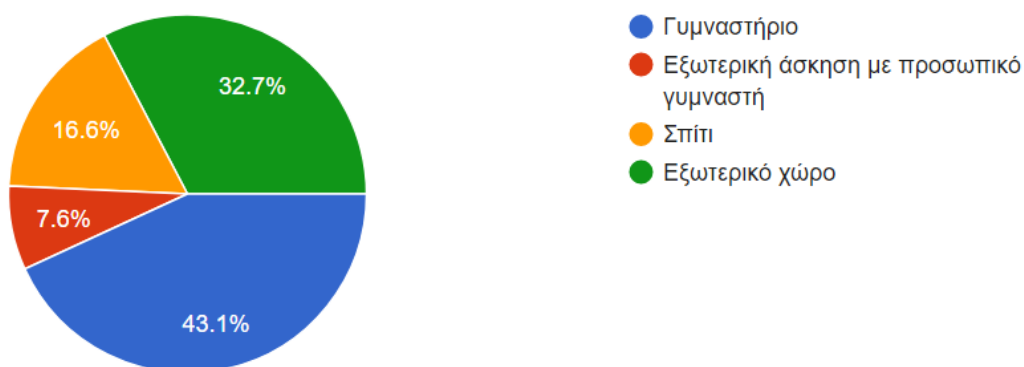
Πριν τον Covid-19 πάνω από το 55% των ερωτηθέντων ασκούσαν σε γυμναστήριο και μόνο το 15% στο σπίτι (Σχήμα 1.5). Κατά τη διάρκεια της καραντίνας το 57.3% επέλεξε να γυμνάζεται στο σπίτι κι ένα σχετικά υψηλό ποσοστό επέλεξε να γυμνάζεται σε εξωτερικό χώρο ατομικά χωρίς γυμναστή (Σχήμα 1.6). Μόνο το 5% επέλεξε να ασκηθεί με την καθοδήγηση προσωπικού γυμναστή. Περισσότερο από το 50% του δείγματος μετά την καραντίνα θα επιλέξει να ασκηθεί εκτός κλειστού γυμναστηρίου, ενώ το 43% θα ασκηθεί σε γυμναστήριο (Σχήμα 1.7).



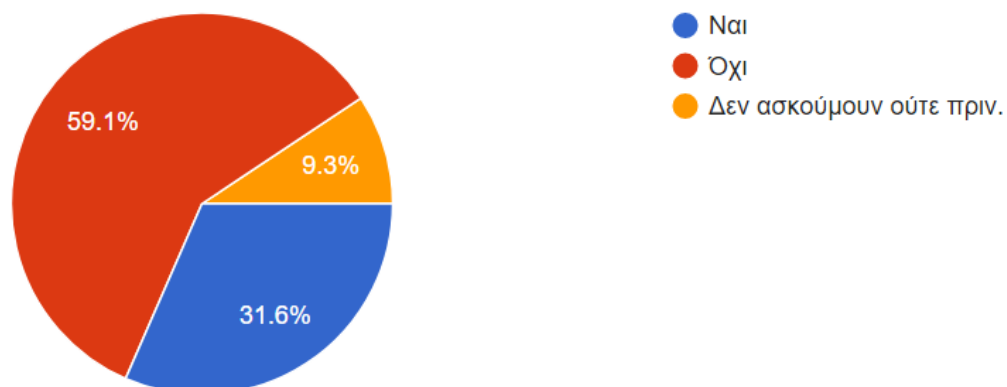
Σχήμα 1.5 Τοποθεσία άσκησης προ Covid-19



Σχήμα 1.6 Τοποθεσία άσκησης κατά την περίοδο καραντίνας Covid-19



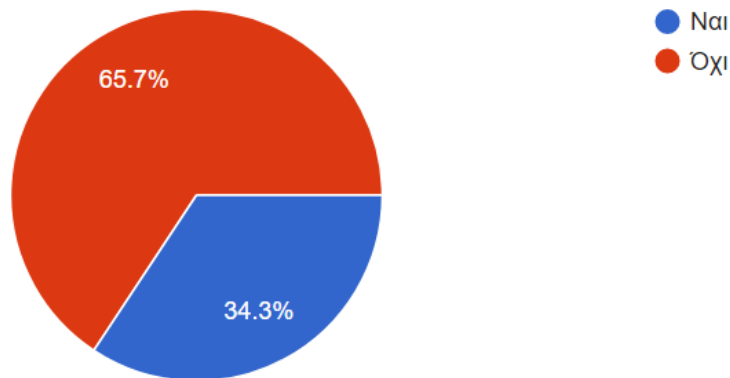
Σχήμα 1.7 Τοποθεσία άσκησης μετά την περίοδο καραντίνας Covid-19



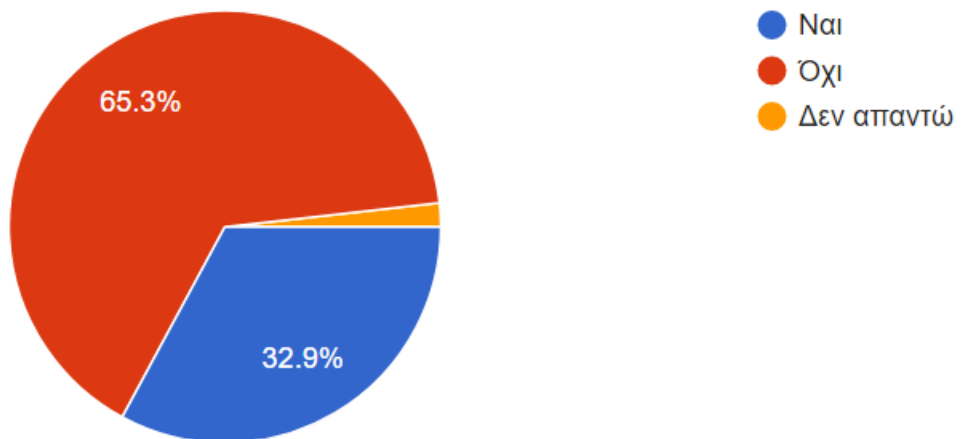
Σχήμα 1.8 Σταμάτησαν τη γυμναστική εν μέσω καραντίνας

Κεφάλαιο 1

Σύμφωνα με την έρευνα, μόνο το 34.3% ασκούταν πιο συχνά στη καραντίνα και το 32.9% ασκήθηκε περισσότερο λόγω του ιού (Σχήματα 1.9 και 1.10).

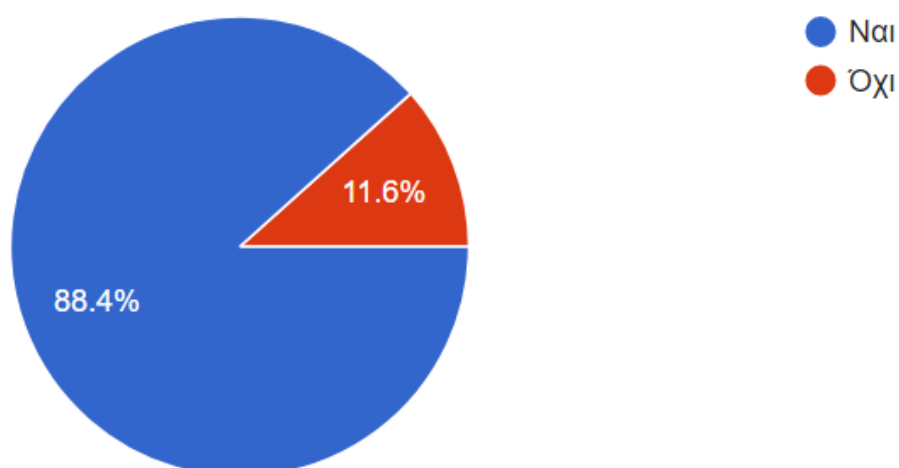


Σχήμα 1.9 Ασκούταν πιο εντατικά στην καραντίνα

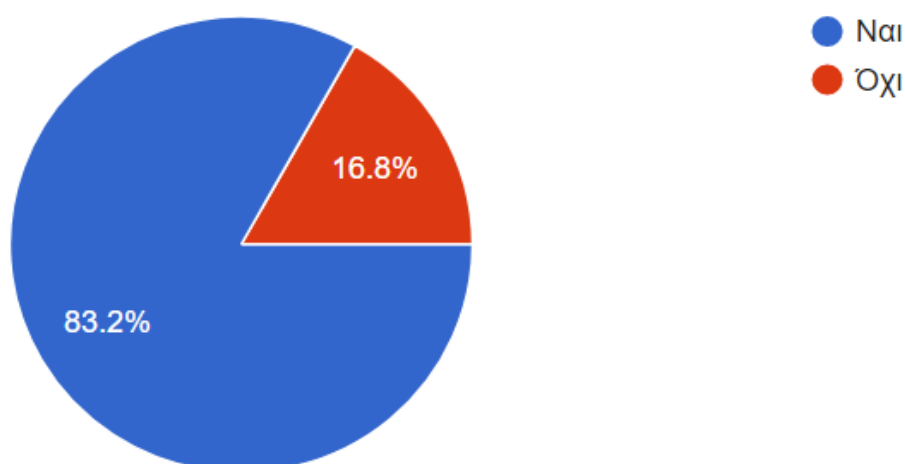


Σχήμα 1.10 Συνέβαλε ο Covid-19 στον να ασκείστε πιο πολύ

Από τα σχήματα 1.11 κι 1.12, παρατηρούμε ότι οι ασκούμενοι ασκούνται και για να βελτιώσουν την υγεία τους αλλά και για να βελτιώσουν την εμφάνισή τους. Είναι μικρό το ποσοστό των ατόμων που ασκούνται και δεν τους νοιάζει η βελτίωσή τους.



Σχήμα 1.11 Άσκηση για βελτίωση υγείας

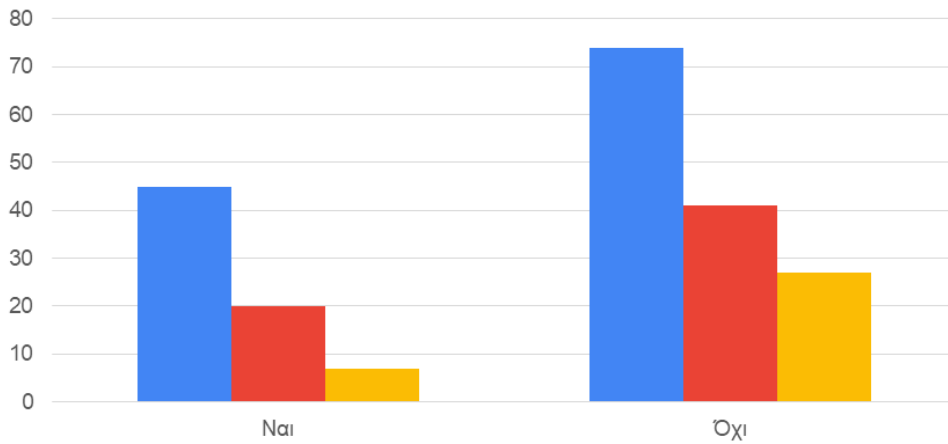


Σχήμα 1.12 Άσκηση για βελτίωση εμφάνισης

1.4 Μελέτη αποτελεσμάτων έρευνας

Αφού άντλησα τα παραπάνω δεδομένα χρησιμοποιώντας το Google Forms, πέρασα τα δεδομένα σε ένα Google Sheet για να καταλήξω σε πιο ειδικά συμπεράσματα συγκρίνοντας τις σχέσεις δεδομένων μεταξύ τους. Ειδικότερα ενδιαφερόμουν να μάθω αν υπάρχει διαφορά στη συχνότητα σωματικής άσκησης μεταξύ αντρών και γυναικών και ηλικίας. Αν οι γυναίκες ή οι άντρες σταμάτησαν να γυμνάζονται κατά τη διάρκεια της καραντίνας ή αν ασκούσαν πιο εντατικά. Το πιο σημαντικό ήταν αν η ηλικιακή ομάδα ή το φύλο έπαιξε ρόλο για να χρησιμοποιήσει κάποιος μια εφαρμογή καταγραφής άσκησης. Τέλος, αν σχετιζόταν η τοποθεσία άσκησης των ατόμων πριν την καραντίνα, με το αν σταμάτησαν να γυμνάζονται. (Σχήμα 1.13)

Χρησιμοποίησα ράβδους για αναπαραστήσω καλύτερα το κάθε γράφημα ανάλογα με το περιεχόμενό τους.



Σχήμα 1.13 Σχέση τοποθεσίας άσκησης προ καραντίνας με την παύση άσκησης

- Γυμναστήριο
- Εξωτερικός χώρος
- Σπίτι

1.5 Στόχοι εφαρμογής

Η εφαρμογή επιδιώκει να δώσει κίνητρο στον άνθρωπο να αθληθεί, με την χρήση της Android συσκευής του και των ειδικών αισθητήρων που αυτή περιέχει για την καταγραφή και καθοδήγηση της σωματικής του άσκησης. Αναλυτικά οι στόχοι της εφαρμογής είναι ο χρήστης :

- Να παρακολουθεί τη σωματική άσκησή του
- Να παρακολουθεί την υγεία του
- Να ακολουθήσει έναν υγιεινό τρόπο ζωής και
- Να ορίζει νέους και να κατακτά προσωπικούς στόχους

1.6 Λειτουργίες Εφαρμογής

Στη Φάση ταυτοποίησης, μετά την έρευνα κατέληξα στις λειτουργίες της εφαρμογής.

- Δημιουργία προφίλ χρήστη
- Καταμέτρηση βημάτων
- Καταμέτρηση χιλιομέτρων που διανύθηκαν
- Καταμέτρηση καμένων θερμίδων
- Καταμέτρηση κατανάλωσης νερού
- Καταμέτρηση σωματικού βάρους
- Εύρεση BMI
- Καθημερινοί ατομικοί στόχοι
- Προβολή χάρτη και εντοπισμός τοποθεσίας
- Προβολή ασκήσεων
- Επεξεργασία προφίλ
- Φόρμα επικοινωνίας με τον προγραμματιστή

1.6.1 Δημιουργία Προφίλ

Για να χρησιμοποιήσει κάποιος την εφαρμογή πρέπει πρώτα να δημιουργήσει προσωπικό λογαριασμό, εισάγοντας τα προσωπικά του στοιχεία όπως και τα στοιχεία σύνδεσης (διεύθυνση ηλεκτρονικού ταχυδρομείου και προσωπικός κωδικός). Αν ο χρήστης διαθέτει λογαριασμό Facebook ή Google, δίνεται η δυνατότητα σύνδεσης με τον αντίστοιχο λογαριασμό αντί να δημιουργήσει νέο λογαριασμό.

1.6.2 Καταμέτρηση βημάτων

Στην Αρχική σελίδα (Home Page Fragment) υπάρχει ένας μετρητής ο οποίος μετράει και δείχνει πόσα βήματα έχει κάνει ο χρήστης ημερησίως. Στη συνέχεια θα αναλυθεί ο τρόπος εύρεσης και καταμέτρησης βημάτων.

1.6.3 Καταμέτρηση χιλιομέτρων που διανύθηκαν

Στην Αρχική σελίδα (Home Page Fragment) υπάρχει ένας μετρητής ο οποίος μετράει και δείχνει πόσα χιλιόμετρα έχει διανύσει ο χρήστης ημερησίως. Στη συνέχεια θα αναλυθεί ο τρόπος υπολογισμού και καταμέτρησης χιλιομέτρων.

1.6.4 Καταμέτρηση καμένων θερμίδων

Στην Αρχική σελίδα (Home Page Fragment) υπάρχει ένας μετρητής ο οποίος μετράει και δείχνει πόσα βήματα έχει κάνει ο χρήστης ημερησίως. Στη συνέχεια θα αναλυθεί ο τρόπος εύρεσης και καταμέτρησης θερμίδων.

1.6.5 Καταμέτρηση κατανάλωσης νερού

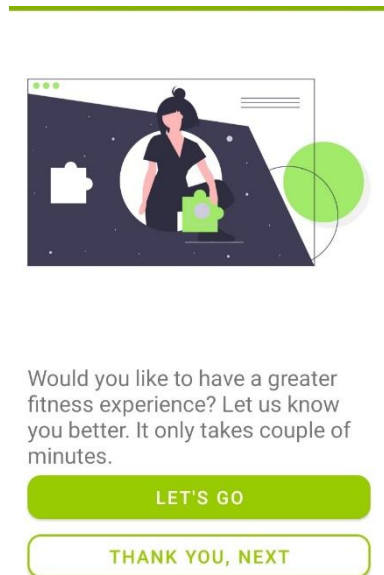
Το νερό είναι εξίσου σημαντικό με την σωματική άσκηση, για την υγεία μας. Έτσι στην Αρχική σελίδα (Home Page Fragment), υπάρχει η δυνατότητα- μετρητής κατανάλωσης νερού. Σύμφωνα με έρευνες [1] ο μέσος άνθρωπος πρέπει να καταναλώνει 2 λίτρα νερό την ημέρα. Στην εφαρμογή ο χρήστης αρχικά έχει ως στόχο τα 2 λίτρα και μπορεί να προσθέσει τα ποτήρια νερού που καταναλώνει.

1.6.6 Καταμέτρηση σωματικού βάρους

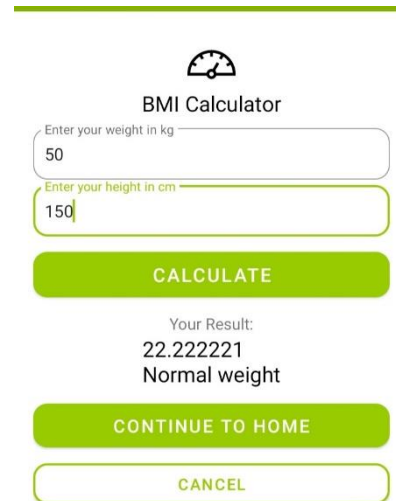
Μετά την δημιουργία νέου λογαριασμού και σύνδεσης στην εφαρμογή για πρώτη φορά, προτείνεται η καταχώρηση βάρους. Αυτό μπορεί να επιτευχθεί και στο Home Page αλλά και στην επιλογή του Προφίλ στο Drawer Shift- Side Menu. Η καταμέτρηση σωματικού βάρους μπορεί να δώσει κίνητρο στον χρήστη να φτάσει στα φυσιολογικά για εκείνον κιλά και να τα ελέγχει.

1.6.7 Εύρεση BMI

Μετά την δημιουργία νέου λογαριασμού και σύνδεσης στην εφαρμογή για πρώτη φορά, προτείνεται (Εικόνα 1.1) η καταχώρηση βάρους και ύψους για τον υπολογισμό του δείκτη μάζας σώματος- BMI. Ο υπολογισμός γίνεται με βάση την εξίσωση: $BMI = \text{βάρος} / (\text{ύψος} * \text{ύψος})$ [2]. Έπειτα ο χρήστης βλέπει κατά πόσο το βάρος του είναι φυσιολογικό, ανάλογα με τη τιμή του BMI (Εικόνα 1.2).



Εικόνα 1.1 Πρόταση Καταχώρησης Βάρους και Ύψους



Εικόνα 1.2 Διεπαφή Υπολογισμού BMI

1.6.8 Θέση και επίτευξη στόχων

Στην εφαρμογή αρχικά υπάρχουν ενδεικτικοί στόχοι, όπως ημερήσιος στόχος βημάτων και κατανάλωσης νερού, οι οποίοι μπορούν να αλλάξουν. Ο χρήστης μπορεί να θέσει νέους στόχους ανάλογα με τις ανάγκες του.

1.6.9 Προβολή χάρτη και εντοπισμός τοποθεσίας

Κατά τις Αερόβιες ασκήσεις, τρέξιμο, περπάτημα, τζόκινγκ και ποδήλατο, ο χρήστης μπορεί να ενεργοποιήσει ένα χρονόμετρο για να καταγράψει τον χρόνο της άσκησης του αλλά και να εντοπίσει την τοποθεσία του στον χάρτη και να δει τις πιθανές διαδρομές.

1.6.10 Προβολή ασκήσεων

Στη παρούσα έκδοση υπάρχουν εικόνες με σιλουέτες στις οποίες προβάλλονται οι ασκήσεις διατάσεων, για προθέρμανση και αποθεραπεία. Στην επόμενη έκδοση στόχος μου είναι να προσθέσω 3D animation με φιγούρες οι οποίες θα δείχνουν τις ασκήσεις, κι έπειτα με την βοήθεια αισθητήρων θα ελέγχεται αν ο χρήστης κάνει σωστά την άσκηση.

1.7 Επίλογος

Η εφαρμογή βοηθάει τον χρήστη στο να φροντίζει τον εαυτό του και να πετυχαίνει τους στόχους του όσον αφορά το σώμα. Στόχος της χρήσης της εφαρμογής είναι να δώσει κίνητρο στον άνθρωπο να ασκηθεί. Δίνει τη δυνατότητα στον χρήστη:

- Να παρακολουθεί την υγεία και την άσκησή του
- Να ακολουθεί υγιεινό τρόπο ζωής
- Να θέτει και να επιτυγχάνει στόχους

Κεφάλαιο 2ο: Τεχνολογίες που χρησιμοποιήθηκαν

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψουν οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της Android εφαρμογής. Επιγραμματικά, η ανάπτυξη έγινε στην πλατφόρμα Android Studio (v. 4), η βάση δεδομένων σχεδιάστηκε στην Firebase, χρησιμοποιήθηκαν αισθητήρες καταγραφής θέσης και κίνησης, προστέθηκε η βιβλιοθήκη Google Maps για την εισαγωγή χαρτών και το Facebook SDK για την διευκόλυνση εισόδου των χρηστών καθώς και τέλος, εφαρμόστηκαν μέθοδοι UX/UI για να προσφέρεται καλύτερη εμπειρία χρήσης.

2.1 Android Studio

Το κύριο μέλημα της πλατφόρμας Android είναι η διευκόλυνση ανάπτυξης νέων εφαρμογών οι οποίες θα είναι για συγκεκριμένες ανάγκες του τελικού χρήστη. Ένα άλλο σημαντικό σημείο είναι η δημιουργία cross platform εφαρμογών, δηλαδή να μπορεί να τρέχουν σε πολλές διαφορετικές συσκευές με διαφορετικό υλικό η καθεμία. Αυτό επιτυγχάνεται με την εξάπλωση και την ευρεία χρήση των Android συσκευών. Η γλώσσα στην οποία συντάχθηκαν οι περισσότερες εφαρμογές είναι η JAVA η οποία μπορεί να εκτελείται σε πολλές συσκευές. Από το 2017 το Android Studio επιτρέπει την ανάπτυξη λογισμικού για Android συσκευές και στη γλώσσα Kotlin της JetBrains [3]. Αυτές οι εφαρμογές είναι στο πιο υψηλό layer του android stack.

2.2 Λειτουργικό Android

Το Android είναι ένα λειτουργικό σύστημα ανοικτού κώδικα για κινητές συσκευές και βασίζεται σε πυρήνα Linux. Τα τελευταία χρόνια έχει αυξηθεί σημαντικά ο αριθμός των χρηστών Android [4] καθώς και η ποικιλία συσκευών με Android, όπως τηλεοράσεις, έξυπνα ρολόγια, ακόμα και έξυπνες ηλεκτρικές σκούπες .

Το σύστημα εκκινεί μια διαδικασία για μια εφαρμογή Android όταν πρέπει να εκτελεστεί μέρος του κώδικά της και θα παραμείνει σε λειτουργία έως ότου δεν χρειάζεται πλέον και το σύστημα πρέπει να ανακτήσει τη μνήμη του για χρήση από άλλες εφαρμογές.

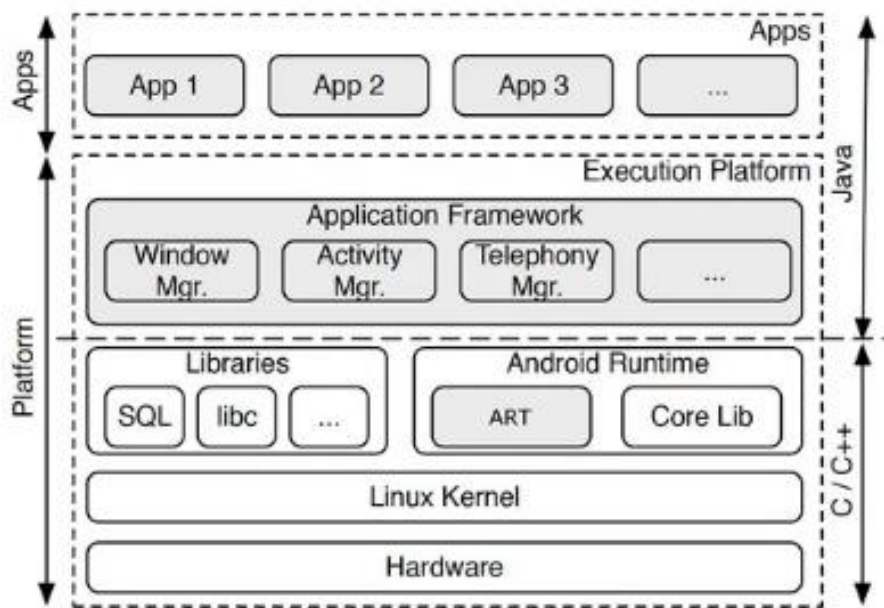
Στο Android η διάρκεια ζωής ενός Activity δεν ελέγχεται άμεσα από την ίδια την εφαρμογή αλλά, καθορίζεται και διαχειρίζεται από το σύστημα ανάλογα του πόσο σημαντική είναι, πόση μνήμη καταλαμβάνει και πόση ελεύθερη μνήμη υπάρχει.

2.3 Αρχιτεκτονική Android

Το Android έχει μια αρχιτεκτονική η οποία περιλαμβάνει τέσσερα επίπεδα που φαίνονται στο Σχήμα 2.1.

1. Εφαρμογή
2. Framework εφαρμογής
3. Android Runtime και βιβλιοθήκες
4. Πυρήνας Linux

Οι εφαρμογές που υπάρχουν στο λειτουργικό Android, επικοινωνούν με ένα application layer το οποίο το οποίο περιλαμβάνει άλλες εφαρμογές. Αυτές οι εφαρμογές είναι γενικού σκοπού και οι οποίες χρησιμοποιούνται από άλλες εφαρμογές.



Σχήμα 2.1 Αρχιτεκτονική Android Συστήματος

1. Επίπεδο εφαρμογής:

Είναι το πρώτο, ξεκινώντας από πάνω, στρώμα της αρχιτεκτονικής του Android και είναι το επίπεδο που χρησιμοποιείται για την εγκατεστημένη εφαρμογή (π.χ. τηλέφωνο, κάμερα κ.λπ.). Οι περισσότερες εφαρμογές είναι κοινές για όλα τα κινητά όπως χάρτες google, κάμερα, web browser, μηνύματα, επαφές που είναι προεγκατεστημένες. Ο τελικός χρήστης χρησιμοποιεί το πλαίσιο εφαρμογής για τη λειτουργία αυτών των εφαρμογών.

2. Application Framework:

Το Application Framework επίπεδο προσφέρει διαφορετικά πακέτα εφαρμογών υπηρεσιών. Περιλαμβάνει απαιτούμενες τάξεις και υπηρεσίες για εφαρμογές Android. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν και να διευρύνουν τα στοιχεία που υπάρχουν ήδη στη διεπαφή προγραμματισμού εφαρμογών. Οι Managers που υπάρχουν σε αυτό το επίπεδο επιτρέπουν την πρόσβαση δεδομένων στην εφαρμογή. Οι Managers αυτοί είναι οι εξής:

- **Activity Manager:** Ο ρόλος του είναι να γίνεται σωστός έλεγχος και σωστή διαχείριση όλων των Activity. Είναι ο υπεύθυνος Managers για τον κύκλο ζωής των εφαρμογών.
- **Resource Manager:** Κάνει δυνατή την πρόσβαση σε πόρους χωρίς κώδικα όπως γραφικά κ.λπ.
- **Notification Manager:** Επιτρέπει σε όλες τις εφαρμογές την εμφάνιση προσαρμοσμένων ειδοποιήσεων στη γραμμή κατάστασης.
- **Location Manager:** Όταν ο χρήστης εισέρχεται ή αποχωρεί από μια συγκεκριμένη γεωγραφική τοποθεσία, ενεργοποιεί ειδοποιήσεις.
- **Package Manager:** Είναι υπεύθυνος για την ανάκτηση δεδομένων σχετικά με τα εγκατεστημένα πακέτα στη συσκευή
- **Telephony Manager:** Διαχειρίζεται τις ρυθμίσεις σύνδεσης δικτύου και όλες τις πληροφορίες που αφορούν τις υπηρεσίες δικτύου.
- **Window Manager:** Δημιουργία προβολών και διατάξεων.
- **Content Manager:** Ενεργοποίηση πρόσβασης σε εφαρμογές δεδομένων από άλλες εφαρμογές, δηλ. Κοινή χρήση δεδομένων [5].

3. Android Runtime και βιβλιοθήκες:

Στο επίπεδο Android Runtime περιέχεται το βασικό στοιχείο ART, το οποίο αντικατέστησε το Dalvik virtual machine (DVM). Στο DVM κάθε διαδικασία εκτελείται σε ξεχωριστή παρουσία στο Virtual Machine, ενώ το ART είναι εξοπλισμένο με έναν Ahead-of-Time compiler. Κατά τη φάση εγκατάστασης της εφαρμογής, μεταφράζει στατικά τον bytecode DEX σε κώδικα μηχανής και το αποθηκεύει στον χώρο αποθήκευσης της συσκευής. Αυτό συμβαίνει μόνο μια φορά κατά την εγκατάσταση της εφαρμογής κι όχι στο runtime, σε αντίθεση με το DVM. Συνεπώς οι εφαρμογές μπορεί να χρειάζονται περισσότερη ώρα να εγκατασταθούν με το ART αλλά τρέχουν πιο γρήγορα κατά τη χρήση τους, χρησιμοποιούν λιγότερη μνήμη και εξοικονομούν μπαταρία. Τέλος, σε αυτό το επίπεδο υπάρχουν πολλές βιβλιοθήκες όπως οι SSL, SQLite ή libc.

4. Πυρήνας Linux:

Στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας της αρχιτεκτονικής του Android, ο οποίος παρέχει σταθερό και αξιόπιστο λειτουργικό σύστημα για φορητές πλατφόρμες και προσφέρει υπηρεσίες όπως διαχείριση μνήμης, διαχείριση μπαταρίας, ασφάλεια κ.λπ. Ο πυρήνας είναι αυτός που παρέχει την επικοινωνία μεταξύ του λογισμικού και του υλικού. Κάνει το σύστημα ασφαλές προσφέροντας βασικές δυνατότητες ασφαλείας όπως:

- Μοντέλο δικαιωμάτων, το οποίο ορίζεται από τον χρήστη
- Απομόνωση διαδικασιών
- Δυνατότητα διαγραφής περιττού και μη ασφαλούς μέρους του πυρήνα

Όταν ένας multiuser χρησιμοποιεί το λειτουργικό σύστημα, ο κύριος σκοπός ασφαλείας του Linux Kernel είναι ο διαχωρισμός των πόρων χρήστη μεταξύ τους. Έτσι:

- Δεν είναι δυνατή η ανάγνωση των δεδομένων άλλου πόρου
- Δεν μπορεί να χρησιμοποιήσει το χώρο μνήμης άλλου πόρου
- Δεν μπορεί να χρησιμοποιήσει τους πόρους της CPU άλλου πόρου
- Δεν μπορεί να χρησιμοποιήσει τις συσκευές άλλου πόρου, όπως GPS, τηλεφωνία, Bluetooth.

2.3.1 Activity Manager

Όπως προαναφέρθηκε, ο συγκεκριμένος Manager είναι υπεύθυνος για τον κύκλο ζωής των διάφορων Activities. Τα Activities στο Android είναι σαν κάθε ξεχωριστό παράθυρο σε μια desktop εφαρμογή. Μια εφαρμογή μπορεί να έχει ένα ή πολλά περισσότερα Activities. Για κάθε ένα Activity υπάρχει ένα κύκλος ζωής και κάποια στάδια από τα οποία περνάει. Για τον κύκλο ζωής των Activities θα αναφερθούμε παρακάτω. Ο Activity Manager διαχειρίζεται τις αλλαγές στα στάδια της ζωής ενός Activity. Με αυτόν τον τρόπο ένας χρήστης μπορεί να εναλλάσσεται μεταξύ των διάφορων εφαρμογών και να μπορεί το λειτουργικό να το αντιλαμβάνεται βοηθώντας σε κρίσιμα ζητήματα τις εφαρμογές, όπως να σταματάει την λειτουργία ενός Activity αν δεν υπάρχει αρκετή μνήμη. Ένα από αυτά τα ζητήματα για παράδειγμα είναι το να μην σταματάει να λειτουργεί-κολλάει μια εφαρμογή όταν αλλάζει ο προσανατολισμός από portrait σε landscape και το αντίστροφο ή όταν συμβαίνει εναλλαγή από μια εφαρμογή σε κάποια άλλη.

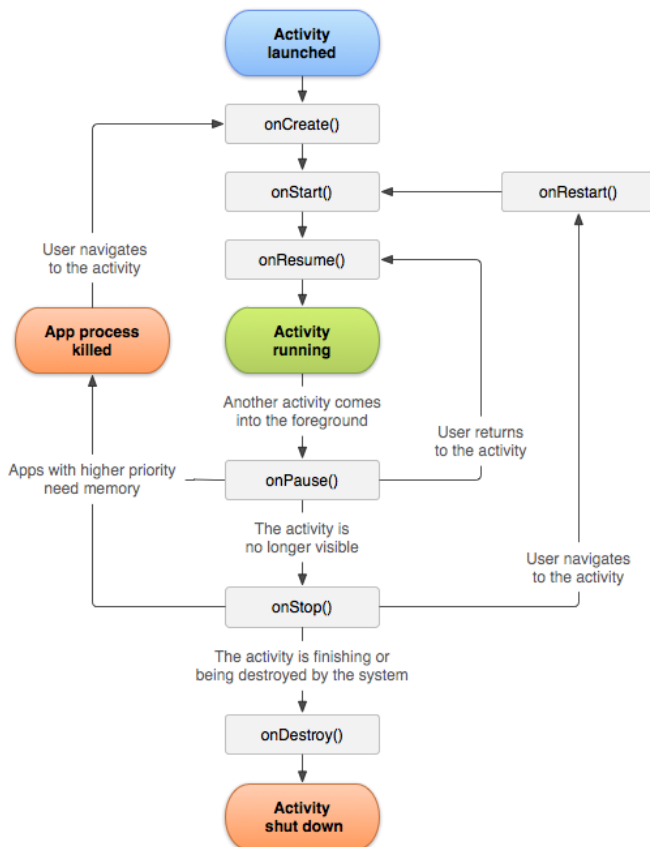
2.4 Activities και Fragments

Activity

Ένα Activity παρέχει το παράθυρο στο οποίο η εφαρμογή αντλεί το UI του. Αυτό το παράθυρο συνήθως γεμίζει την οθόνη, αλλά μπορεί να είναι μικρότερο από την οθόνη και να αιωρείται πάνω από άλλα παράθυρα. Γενικά, ένα Activity εφαρμόζει μία οθόνη σε μια εφαρμογή. Ένα Activity μπορεί να λειτουργεί αυτόνομα αλλά μπορεί σε αυτό να φορτώνονται άλλα Fragments.

Οι περισσότερες εφαρμογές περιέχουν πολλές οθόνες, πράγμα που σημαίνει ότι περιλαμβάνουν πολλά activities. Συνήθως, ένα activity σε μια εφαρμογή ορίζεται ως το κύριο activity, το οποίο είναι η πρώτη οθόνη που εμφανίζεται όταν ο χρήστης ξεκινά την εφαρμογή. Στην εφαρμογή μου το activity που έχει δηλωθεί ως το κύριο, είναι το Home Activity.

Κάθε activity μπορεί στη συνέχεια να ξεκινήσει ένα άλλο activity για να εκτελέσει διαφορετικές ενέργειες. Για παράδειγμα, το κύριο activity σε μια απλή εφαρμογή ηλεκτρονικού ταχυδρομείου μπορεί να παρέχει την οθόνη που εμφανίζει τα εισερχόμενα e-mail. Από εκεί, το κύριο activity μπορεί να ξεκινήσει άλλα activities που παρέχουν οθόνες για εργασίες όπως η σύνταξη e-mail και το άνοιγμα μεμονωμένων e-mail. Στο Σχήμα 2.2 περιγράφεται ο κύκλος ζωής ενός Activity.

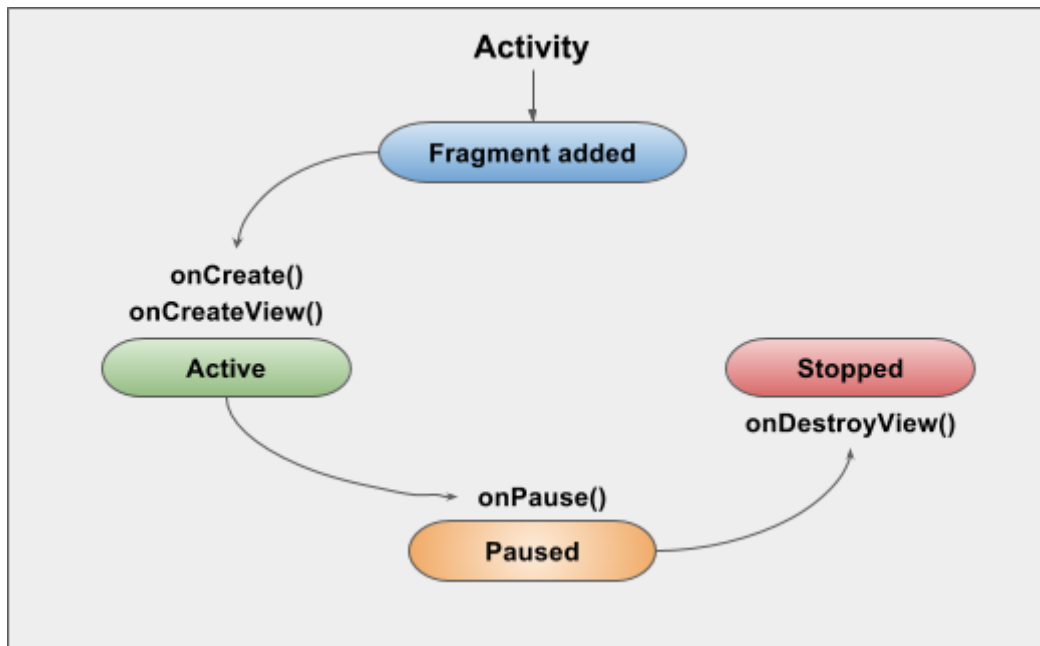


Σχήμα 2.2 Activity Lifecycle [27]

Fragments

Ένα Fragment είναι ένα μέρος ενός Activity που κάνει επιτρεπτό έναν πιο αρθρωτό και καθαρό σχεδιασμό για το Activity. Ένα Fragment θα μπορούσε να χαρακτηριστεί κι ως ένα είδος υπο-activity [28]. Τα πιο σημαντικά στοιχεία των Fragments είναι:

- Ένα Fragment έχει τη δική του διάταξη και τη δική του συμπεριφορά με τα δικά του lifecycle callbacks (Σχήμα 2.3).
- Μπορούν να προστεθούν και να αφαιρεθούν Fragments σε ένα κατά τη διάρκεια εκτέλεσης του Activity.
- Μπορεί να δημιουργηθεί ένα περιβάλλον εργασίας πολλών παραθύρων, συνδυάζοντας πολλαπλά Fragments σε ένα Activity.
- Ένα Fragment μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά Activity.
- Ο κύκλος ζωής ενός Fragments είναι στενά συνδεδεμένος με τον κύκλο ζωής του host activity. Δηλαδή όταν το activity σταματήσει να εκτελείται, όλα τα Fragments που διατίθενται στο Activity θα σταματήσουν να εκτελούνται κι αυτά.
- Ένα Fragment μπορεί να εφαρμόσει μια συμπεριφορά που δεν έχει interface component
- Τα Fragments προστέθηκαν στο Android API στην Honeycomb έκδοση, το οποίο είναι η έκδοση 11 του API.



Σχήμα 2.3 Κύκλος ζωής Fragment

Διαφορές Activities με Fragments

Τα Activities και τα Fragments έχουν κάποιες διαφορές από τις οποίες εξαρτάται ποιο είναι καταλληλότερο να χρησιμοποιηθεί σε κάθε διεπαφή. [Πίνακας 2.1]

Μια κύρια διαφορά είναι ότι ένα Activity μπορεί να υπάρξει χωρίς Fragment. Ένα Fragment όμως δεν μπορεί να σταθεί μόνο του χωρίς να καλείται από ένα ή περισσότερα Activity. Ένα Fragment αντιπροσωπεύει μια συμπεριφορά ή ένα τμήμα του User Interface σε ένα Activity.

Πίνακας 2.1 Διαφορές χαρακτηριστικών Activity-Fragment [29]

Χαρακτηριστικά	Activity	Fragment
Lightweight	OXI	NAI
Reusable	OXI	NAI
App entry point	NAI	OXI
Easy to manage navigation	NAI	OXI

Όσον αφορά τον κώδικα, με διαφορετικό τρόπο καλεί και καλείται ένα Activity και με διαφορετικό ένα Fragment. Όπως επίσης όταν θέλουμε να αναφερθούμε στο αντικείμενο ενός Activity, μπορούμε να χρησιμοποιήσουμε το “this”, ενώ για το αντικείμενο ενός Fragment πρέπει να καλέσουμε τη μέθοδο getActivity() (Πίνακας 2.2). Μια ακόμα διαφορά είναι η εισαγωγή περιεχομένου σε ένα αντικείμενο που δημιουργήθηκε στον Java κώδικα και το casting του από κάποιο αντικείμενο xml. (Πίνακας 2.3)

Πίνακας 2.2 Διαφορές Intent Activity-Fragment

Activity	<pre>Intent intentContact = new Intent(packageContext: HomeActivity.this, ContactUsActivi startActivity(intentContact);</pre>
Fragment	<pre>startActivity(new Intent(getActivity(),WaterActivity.class));</pre>

Πίνακας 2.3 Διαφορές Casting Activity-Fragment

Activity	<pre>drinkBtn = (Button) findViewById(R.id.drinkBtn); decGlassBtn = (Button) findViewById(R.id.decGlassBtn); glassCounter=(TextView) findViewById(R.id.glassCounter);</pre>
Fragment	<pre>username=(TextView) view.findViewById(R.id.usernameField); progressBar = (ProgressBar) view.findViewById(R.id.progressBar); steps=(TextView) view.findViewById(R.id.steps);</pre>

2.5 Αισθητήρες

2.5.1 Αισθητήρες κίνησης

Το Android διαθέτει διάφορους αισθητήρες που μπορούν να καταγράφουν τη κίνηση μιας συσκευής. Κάποιοι είναι software-based και κάποιοι άλλοι hardware-based. Οι software-based αισθητήρες αντλούν τα δεδομένα τους από έναν ή περισσότερους hardware-based. (πχ είτε από το επιταχυνσιόμετρο και το μαγνητόμετρο είτε από το γυροσκόπιο). Οι αισθητήρες κίνησης χρησιμοποιούνται για την παρακολούθηση των κινήσεων της συσκευής. Οι κινήσεις που μπορούν να καταγραφούν είναι η κλίση, το κούνημα, η περιστροφή αλλά και η ταλάντευση.

Η κίνηση είναι συνήθως μια αντανάκλαση της άμεσης εισόδου χρήστη (για παράδειγμα, ένας χρήστης που οδηγεί ένα αυτοκίνητο σε ένα παιχνίδι ή ένας χρήστης που ελέγχει μια μπάλα σε ένα παιχνίδι), αλλά μπορεί επίσης να αντικατοπτρίζει το φυσικό περιβάλλον στο οποίο κάθεται η συσκευή. (για παράδειγμα, μετακίνηση μαζί σας ενώ οδηγείτε το αυτοκίνητό σας).

Στην πρώτη περίπτωση, παρακολουθείτε την κίνηση σε σχέση με το πλαίσιο αναφοράς της συσκευής ή το πλαίσιο αναφοράς της εφαρμογής σας. Στη δεύτερη περίπτωση παρακολουθείτε την κίνηση σε σχέση με το παγκόσμιο πλαίσιο αναφοράς.

2.5.1.1 Επιταχυνσιόμετρο (Accelerometer)

Το επιταχυνσιόμετρο που χρησιμοποιείται στην εφαρμογή για την συλλογή δεδομένων κίνησης, και το οποίο διατίθεται στις περισσότερες συσκευές Android, είναι hardware-based. Γενικά όλοι οι αισθητήρες κίνησης επιστρέφουν πολυδιάστατους πίνακες με τις τιμές που καταγράφει ο κάθε αισθητήρας. Το επιταχυνσιόμετρο συλλέγει δεδομένα επιτάχυνσης στους τρεις άξονες συντεταγμένων.

Ο εναλλακτικός αισθητήρας που θα μπορούσε να χρησιμοποιηθεί είναι το γυροσκόπιο. Το γυροσκόπιο συλλέγει δεδομένα περιστροφής στους 3 άξονες. Είναι πιο ευαίσθητο, γρήγορο και προτιμάται για χρήση σε 3D εφαρμογές.

Επιλέχτηκε η χρήση του επιταχυνσιόμετρο αντί του γυροσκοπίου διότι κάθε συσκευή διαθέτει επιταχυνσιόμετρο ενώ γυροσκόπιο όχι. Επίσης το γυροσκόπιο ξοδεύει περισσότερη μπαταρία.

2.5.2 Αισθητήρας Θέσης

Οι αισθητήρες κίνησης από μόνοι τους δεν χρησιμοποιούνται συνήθως για την παρακολούθηση της θέσης της συσκευής, αλλά μπορούν να χρησιμοποιηθούν με άλλους αισθητήρες, όπως ο αισθητήρας γεωμαγνητικού πεδίου, για τον προσδιορισμό της θέσης μιας συσκευής σε σχέση με το παγκόσμιο πλαίσιο αναφοράς.

2.6 Google Maps JavaScript API

Οι χάρτες της Google (Google Maps) δημιουργήθηκαν το 2005 κι έφεραν την επανάσταση στον Παγκόσμιο Ιστό (WWW) όσον αφορά τις διαδικτυακές εφαρμογές χαρτογράφησης. Βασίζονται στην ασύγχρονη JavaScript και XML (AJAX), για τη διατήρηση μιας συνεχούς σύνδεσης μεταξύ του πελάτη και του διακομιστή για άμεση λήψη επιπλέον πληροφοριών χάρτη [6]

Επιπλέον, η Google παρέχει επίσης στους προγραμματιστές τις εκτενείς πηγές κώδικα που ονομάζεται Application Programming Interface (API). Το API αποτελείται από ένα σύνολο δομών δεδομένων, κλάσεων αντικειμένων ή συναρτήσεων που μπορούν να χρησιμοποιηθούν από έναν προγραμματιστή χρησιμοποιώντας JavaScript, PHP ή άλλη γλώσσα scripting.

Το Google Maps API είναι επίσης καλά τεκμηριωμένο, με πολλά σεμινάρια και δείγματα στο διαδίκτυο για να μελετήσουν οι φοιτητές. Υπάρχει επίσης ένα ενεργό φόρουμ χρηστών για ερωτήσεις και απαντήσεις.

Είναι μια δωρεάν υπηρεσία, αλλά έχει ορισμένους τεχνικούς περιορισμούς. Για παράδειγμα, υπάρχει ένα όριο στον αριθμό των ερωτημάτων που ένας χρήστης μπορεί να υποβάλει στις υπηρεσίες Χαρτών Google ανά ημέρα και ανά δευτερόλεπτο. Οι χρήστες μπορούν να αυξήσουν το όριο με συνδρομές επί πληρωμή, αλλά η συντριπτική πλειονότητα των χρηστών σπάνια φτάνει το προεπιλεγμένο όριο. [7]

Υποστηρίζεται από προγράμματα περιήγησης Ιστού αλλά και σε συσκευές όπως το Apple iPad και iPhone, και φυσικά σε κινητές συσκευές Android. Η Google διαχώρισε τα Google Maps APIs σύμφωνα με τη χρήση του κάθε API.

Αυτές οι δυνατότητες καθιστούν το Google Maps API ως το πιο συχνά χρησιμοποιούμενο API Χαρτών για χαρτογράφηση στο διαδίκτυο σύμφωνα με έρευνα του 2018. [8]

Η Google διαχώρισε τα Google Maps APIs σύμφωνα με τις υπηρεσίες που παρέχει το κάθε API:

- Maps JavaScript API – Παρέχει γενικές πληροφορίες χαρτών και διαδρομών. Αναλύεται περαιτέρω παρακάτω.
- Google Geocoding API - Μετατρέπει διευθύνσεις σε γεωγραφικές συντεταγμένες και αντίστροφα.
- Google Places API - Παρέχει πιο λεπτομερείς πληροφορίες σχετικά με ένα συγκεκριμένο μέρος.
- Google Timezone API - Παρέχει δεδομένα ζώνης ώρας για οποιαδήποτε τοποθεσία.
- Google Maps Distance Matrix API - Υπολογίζει τις αποστάσεις και τους χρόνους ταξιδιού με βάση τα σημεία έναρξης και προορισμού, συμπεριλαμβανομένων επιλογών για διαφορετικούς τρόπους μεταφοράς, όπως οδήγηση, δημόσια συγκοινωνία, περπάτημα ή ποδηλασία.
- Google Maps Driving Directions API - Παρέχει οδηγίες μεταξύ τοποθεσιών, συμπεριλαμβανομένων επιλογών για διαφορετικούς τρόπους μεταφοράς, όπως οδήγηση, δημόσια συγκοινωνία, περπάτημα ή ποδηλασία.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε το Maps JavaScript API, το οποίο κάνει δυνατή την προσαρμογή χαρτών από τον προγραμματιστή, που τον επιτρέπει την προσθαφαίρεση περιεχομένου και εικόνων για προβολή σε ιστοσελίδες και κινητές συσκευές. Το Maps JavaScript API διαθέτει τέσσερις βασικούς τύπους χαρτών (χάρτης πορείας, δορυφόρος, υβριδικό και έδαφος), οι οποίοι μπορούν να τροποποιηθούν χρησιμοποιώντας στρώματα και στυλ, στοιχεία ελέγχου και συμβάντα και διάφορες υπηρεσίες και βιβλιοθήκες.

2.7 Facebook SDK

Στην εφαρμογή μπορούν οι χρήστες να συνδεθούν και με τον Facebook λογαριασμό τους.

Χρησιμοποιήθηκε το Facebook SDK [9] για Android για την ενσωμάτωση της εφαρμογής με το Facebook. Με τη χρήση του Facebook SDK δίνονται οι παρακάτω δυνατότητες:

- Facebook Analytics – Analytics για το πώς χρησιμοποιούν οι χρήστες το προϊόν.
- Σύνδεση στο Facebook - Έλεγχος ταυτότητας ατόμων με τα διαπιστευτήριά τους στο Facebook.
- Κοινή χρήση και αποστολή διαλόγων - Ενεργοποίηση κοινής χρήσης περιεχομένου από την εφαρμογή στο Facebook.
- Εκδηλώσεις εφαρμογών – Καταγραφή συμβάντων (logs) στην εφαρμογή.
- Graph API - Περιήγηση και ανάπτυξη του Graph API.

Για να επιτευχθεί αυτό, είναι αναγκαία η δημιουργία λογαριασμού Facebook Developer για τον προγραμματιστή. Έπειτα πρέπει να προστεθεί το SDK στο build dependency και τέλος να εισαχθεί το SDK στο πρότζεκτ. Η ελάχιστη έκδοση API που απαιτείται για την χρήση του Facebook SDK, είναι το API 15:Android 4:0:3 (IceCreamSandwich). Μετά τη δημιουργία του πρότζεκτ, προστίθεται στο αρχείο build.gradle (Project) το mavenCentral(). Τέλος στο αρχείο build.gradle (module: app), πρέπει να προστεθεί το 'com.facebook.android:facebook-android-sdk:[5,6]' για να μεταγλωττιστεί η τελευταία έκδοση του Facebook SDK.

Όταν χρησιμοποιείται το Facebook SDK, ορισμένα συμβάντα στην εφαρμογή καταγράφονται αυτόματα και συλλέγονται για το Facebook Analytics, εκτός εάν απενεργοποιηθεί η αυτόματη καταγραφή συμβάντων.

2.8 Firebase Firestore

Αποφάσισα να χρησιμοποιήσω την Firebase της Google ως Βάση Δεδομένων. Είναι εύχρηστη και noSQL και διαδικτυακή βάση δεδομένων. Η χρήση της γίνεται κυρίως για τον συγχρονισμό εφαρμογών client-side και server-side.

Τα κύρια στοιχεία του firestore είναι:

- Ευελιξία: Το firestore υποστηρίζει ευέλικτες και ιεραρχικές δομές δεδομένων. Τα έγγραφα μπορούν να περιέχουν σύνθετα και εμφωλευμένα αντικείμενα.
- Εκφραστικά ερωτήματα: Το firestore υποστηρίζει ερωτήματα για την ανάκτηση εγγράφων. Τα ερωτήματα αυτά μπορούν να περιέχουν πολλαπλά και σύνθετα φίλτρα για την ανάκτηση συγκεκριμένης πληροφορίας.
- Ενημερώσεις πραγματικού χρόνου: Το firestore ενημερώνει τα δεδομένα σε όλες τις συνδεδεμένες συσκευές σε πραγματικό χρόνο.
- Υποστήριξη σε περιπτώσεις που δεν υπάρχει δίκτυο: Το firestore αποθηκεύει τα δεδομένα σε κάθε συσκευή, με τέτοιο τρόπο ώστε να λειτουργεί ακόμα και αν δεν υπάρχει σύνδεση στο δίκτυο.

Επίσης στο site της Firebase διατίθενται έτοιμες χρήσιμες μέθοδοι για την κάθε γλώσσα προγραμματισμού που χρησιμοποιείται. Στη εικόνα 2.1 παρουσιάζεται ενδεικτικά η μέθοδος Firebase Authentication για Android συσκευές και συγκεκριμένα στη γλώσσα προγραμματισμού Java. Με το Firebase Authentication γίνεται έλεγχος ταυτότητας με τη Firebase χρησιμοποιώντας τις διευθύνσεις ηλεκτρονικού ταχυδρομείου και τους κωδικούς πρόσβασης και τη διαχείριση των λογαριασμών που βασίζονται στον κωδικό πρόσβασης της εφαρμογής. Χρησιμοποιείται στη διαδικασία Σύνδεσης στην εφαρμογή.

Easy to integrate on iOS, Android, and the Web

Firebase provides detailed documentation and cross-platform SDKs to help you build and ship apps on Android, iOS, the web, C++, and Unity.



Εικόνα 2.1 Firebase methods

2.9 UX/UI

Ο σχεδιασμός εμπειρίας χρήστη (UX) για τα κινητά είναι ένας σημαντικός τομέας. Επικεντρώνεται κυρίως στη συνολική εμπειρία του προϊόντος και όχι μόνο στην εμφάνισή του. Οι εφαρμογές πρέπει να είναι καλαίσθητες, εύχρηστες, κατανοητές και να παρέχουν εξαιρετική εμπειρία αλληλεπίδρασης. Η πλοήγηση πρέπει να είναι εύκολη και ο χρήστης να εντοπίζει γρήγορα αυτό που αναζητεί. [26]

Μια κακή σχεδίαση UX κάνει τις εφαρμογές και τις ιστοσελίδες δύσχρηστες, δυσνόητες, ακόμα και χαοτικές κάποιες φορές. Εάν η εφαρμογή έχει κακή σχεδίαση UX, οι χρήστες θα δυσκολευτούν να τη χρησιμοποιήσουν και θα αναζητήσουν μια καλύτερη, ευκολότερη εφαρμογή. [10]

- Αυξάνει την παραγωγικότητα
- Αυξάνει τις πωλήσεις και τα έσοδα
- Μειώνει το κόστος υποστήριξης και εκπαίδευσης
- Μειώνει το κόστος ανάπτυξης και χρόνου ανάπτυξης
- Μειώνει το κόστος συντήρησης
- Αυξάνει την ικανοποίηση των πελατών

Στόχος μου πέρα από την ανάπτυξη λογισμικού εφαρμογής υγείας, ήταν να ερευνήσω τον τρόπο ώστε να αναπτύξω μια εύχρηστη εφαρμογή για όλους τους χρήστες κινητών, ανεξαρτήτου φύλου, ηλικίας, μόρφωσης, επιπέδου εξοικείωσης με ηλεκτρονικές συσκευές και προβλημάτων όρασης.

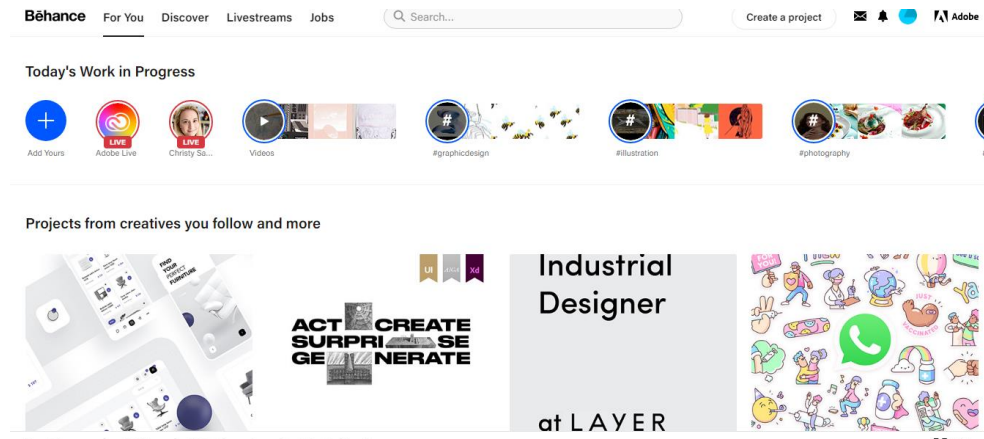
2.9.1 Διαδικασία σχεδίασης UX

Ο σχεδιασμός UX ακολουθεί τη διαδικασία σχεδιασμού με επίκεντρο τον χρήστη. Τα βήματα συνήθως είναι:

- Αναζήτηση και πλάνο
- Στρατηγική
- Έρευνα
- Ανάλυση
- Σχέδιο
- Παραγωγή

2.9.1.1 Αναζήτηση Ιδεών

Για την αναζήτηση ιδεών χρησιμοποιήθηκε το Behance [11] (Εικόνα 2.2), το οποίο είναι μια πλατφόρμα κοινωνικής δικτύωσης που ανήκει στην Adobe και χρησιμοποιείται για προβολή και ανακάλυψη δημιουργικού έργου [12], και άλλες παρόμοιες πλατφόρμες ώστε να ανακαλύψω νέες, σύγχρονες ιδέες σχεδίασης.



Εικόνα 2.2 Behance

2.9.1.2 Έρευνα UX/UI

2.9.1.2.1 Χρώματα και Ψυχολογία Χρωμάτων

Η εφαρμογή που ανέπτυξα θέτει ως στόχο την καθημερινή άσκηση. Έτσι έπειτα από την έρευνα στην ψυχολογία χρωμάτων [13], ανακάλυψα ότι το Πράσινο, το Κίτρινο και το Πορτοκαλί, αποδίδουν ενέργεια και χαρά [14] όπως παρουσιάζεται στον πίνακα 2.4. Οπότε και τα επέλεξα για τα βασικά χρώματα της εφαρμογής μου. Το πράσινο το επέλεξα και για την απόδοση του Eco-friendly.

Πίνακας 2.4 Γενική ψυχολογία χρώματος

Κόκκινο	Κίνδυνος, προσοχή, επίκληση
Λευκό	Καθαρό, παρθένο, νέο
Μπλε	Υγεία, δροσιά
Πράσινο	Φύση, χαλαρωτικό, ροή
Κίτρινο	Ικανότητα, ευτυχία
Πορτοκαλί	Ζεστασιά, ζωντάνια
Μωβ	Αρχοντιά, δύναμη
Μαύρο	Μυστικότητα, αριστοκρατικό, ακριβό

Όσον αφορά το φόντο της εφαρμογής, κατέληξα να σχεδιάσω σε λευκό φόντο. Αν και η «μόδα» είναι η μαύρη οθόνη και γενικά το σκούρο περιβάλλον, σκεφτείτε πόσο δύσχρηστη θα ήταν μια εφαρμογή με μαύρο φόντο κάτω από τον ήλιο της Ελλάδας. Πραγματικά θα ήταν μια «αόρατη εφαρμογή».

2.9.1.2.2 Σχήματα Components

Τα σχήματα μπορούν να χρησιμοποιηθούν για να αντικατοπτρίζουν έναν συγκεκριμένο σκοπό ή νόημα. Το κείμενο ή τα εικονίδια μπορούν να ενισχύσουν αυτό το νόημα, όταν το σχήμα μιας επιφάνειας μπορεί να είναι διφορούμενο. [15] Για παράδειγμα, τα σχήματα κύκλου, οβάλ και έλλειψης δημιουργούν ένα θετικό συναισθηματικό μήνυμα. Η έλλειψη γωνιών των σχημάτων τα καθιστά πιο ευχάριστα και φιλικά. [16] [34]

2.9.2 Σχεδιασμός Διεπαφών

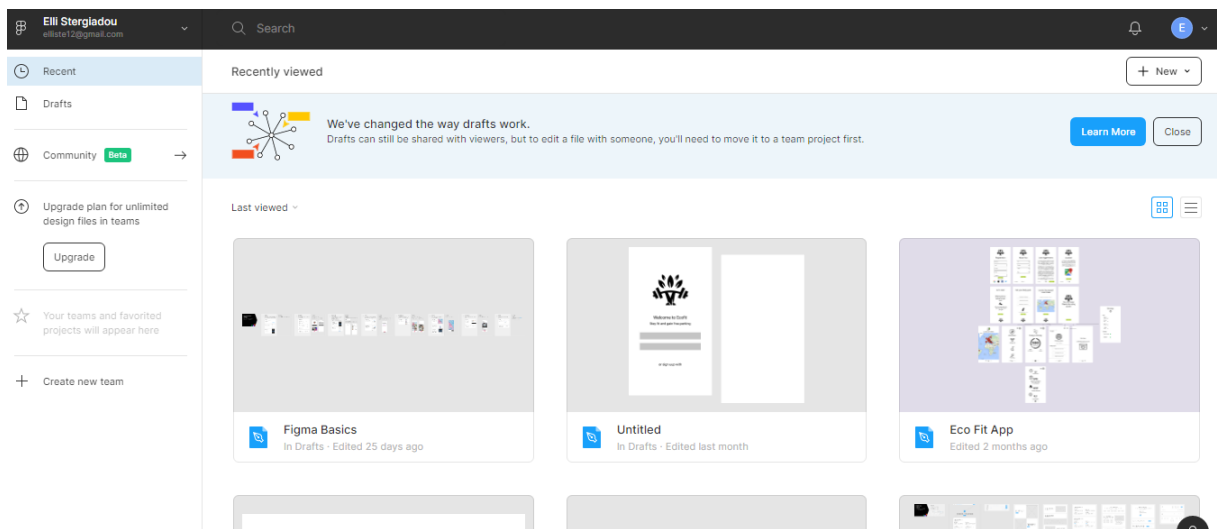
Για τον σχεδιασμό των διεπαφών υπάρχουν δύο κατηγορίες σχεδίασης πρωτοτύπου, η Low-fidelity (lo-fi) και η High-fidelity (hi-fi). Η lo-fi πρωτοτυποποίηση θεωρείται ότι είναι ο ευκολότερος και γρηγορότερος τρόπος [32] για να σχεδιαστεί ένα πρωτότυπο του προϊόντος, συνήθως γίνεται στο χαρτί (πχ. κάρτες) και όπως είναι λογικό, δεν υπάρχουν αλληλεπιδράσεις (Εικόνα 2.3). Η hi-fi πρωτοτυποποίηση είναι μια πιο αργή διαδικασία αλλά η σχεδίαση είναι πολύ κοντά στο τελικό προϊόν [33], κάτι που ίσως κάνει ταχύτερη την ανάπτυξη του κώδικα των διεπαφών της εφαρμογής. Σχεδιάζονται wireframes για κάθε οθόνη και users flows. Επίσης προστίθενται αλληλεπιδράσεις, χρώματα, γραμματοσειρές, εικόνες και τυχόν γραφικά που θα χρησιμοποιηθούν στις διεπαφές.



Εικόνα 2.3 Lo-fi Storyboard [39]

Στην παρούσα εργασία χρησιμοποιήθηκε η hi-fi πρωτοτυποποίηση για την σχεδίαση διεπαφών με την δημιουργία wireframes. Το wireframing είναι μια διαδικασία κατά την οποία σχεδιάζεται χονδρικά το περιεχόμενο και οι οπτικές λεπτομέρειες των εφαρμογών/ ιστοτόπων. Τα Wireframes θεωρούνται «γνωστικό εργαλείο» που βοηθούν στη δημιουργία μιας ιδέας νωρίς στη διαδικασία σχεδιασμού του ιστότοπου ή αντίστοιχα μιας εφαρμογής, και χρησιμοποιούνται ως τρόπος οργάνωσης του περιεχομένου και της βασικής εμφάνισης για να δημιουργήσουν μια κοινή κατανόηση του πώς θα μοιάζει μια εφαρμογή. Μειώνουν την πιθανότητα σφαλμάτων που προέκυψαν κατά την πραγματική διαδικασία ανάπτυξης εφαρμογής, αυτό έχει ως αποτέλεσμα και τη μείωση χρόνου κατασκευής της εφαρμογής.

Για την κατασκευή των wireframes στην παρούσα εφαρμογή χρησιμοποιήθηκε μια πλατφόρμα-εργαλείο σχεδίασης το οποίο ονομάζεται Figma[17]. Το Figma είναι ένα διανυσματικό πρόγραμμα επεξεργασίας γραφικών και ένα πρωτότυπο εργαλείο το οποίο είναι κατά κύριο λόγο web-based, με πρόσθετες δυνατότητες εκτός σύνδεσης που ενεργοποιούνται από εφαρμογές για υπολογιστές MacOS και Windows. Οι συνοδευτικές εφαρμογές Figma Mirror για Android και iOS επιτρέπουν την προβολή πρωτοτύπων Figma σε πραγματικό χρόνο σε κινητές συσκευές. Στο Figma κάνει δυνατό τον διαμοιρασμό των wireframes μεταξύ ατόμων μιας ομάδας. Κάθε άτομο μπορεί να δουλεύει σε πολλά διαφορετικά πρότζεκτς όπως παρουσιάζεται στην εικόνα 2.4. Όσον αφορά την φάση σχεδίαση των wireframes που κατασκευάστηκαν για την εφαρμογή της πτυχιακής, θα αναφερθούμε αναλυτικότερα στην ενότητα 3.3.

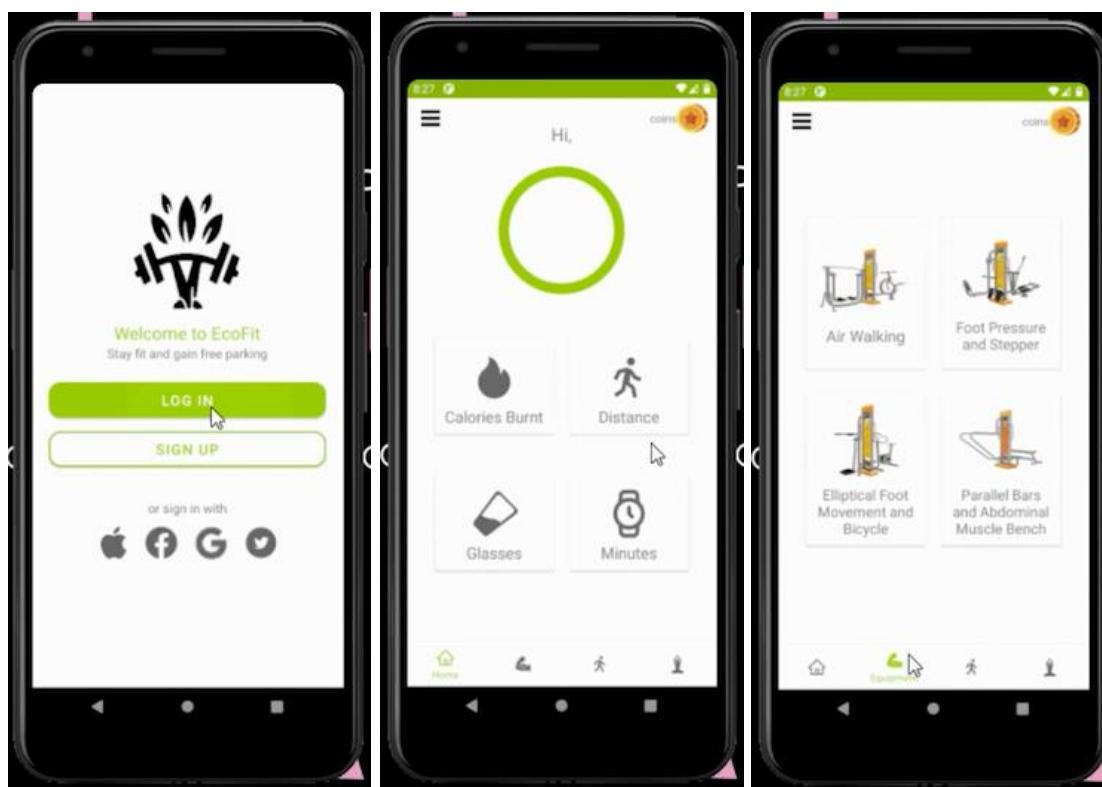


Εικόνα 2.4 Figma

Κεφάλαιο 3ο: Η εφαρμογή

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται η εφαρμογή που ανέπτυξα στο Android Studio. Στο τέλος του κεφαλαίου υπάρχει το ένα βίντεο record που δείχνει πως λειτουργεί η εφαρμογή. Στην πρώτη έκδοση κατασκεύασα μια εφαρμογή που θα εξυπηρετούσε ένα οικολογικό «πράσινο» πάρκο με υπόγειο χώρο στάθμευσης. Στην Εικόνα 3.1 και στο βίντεο 1 παρουσιάζεται αυτή η έκδοση. Στην έκδοση που παρουσιάζεται στην παρούσα πτυχιακή, έχουν αφαιρεθεί οι πόντους που αυξάνονται κατά την άσκηση, που είχαν σκοπό την επιβράβευση με δωρεάν χρόνο στάθμευσης. Επίσης αφαιρέθηκαν το Fragment με τα όργανα γυμναστικής, και τις οδηγίες τους, που θα ήταν τοποθετημένα στα πάρκα.



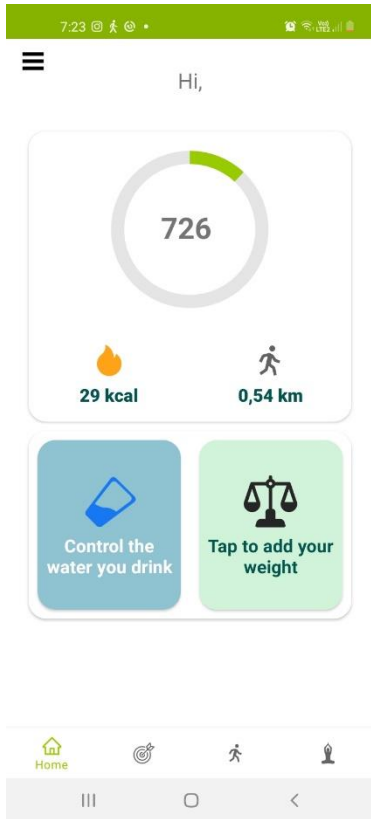
Εικόνα 3.1 Διεπαφές πρώτης έκδοσης της εφαρμογής



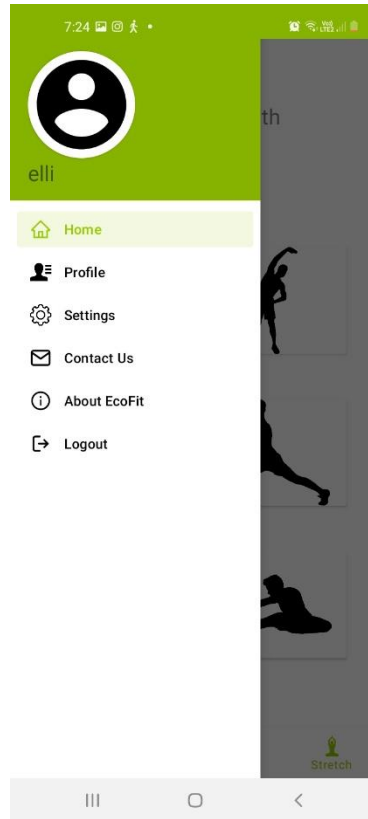
Video.mp4

Βίντεο 1 Πρώτη έκδοση εφαρμογής

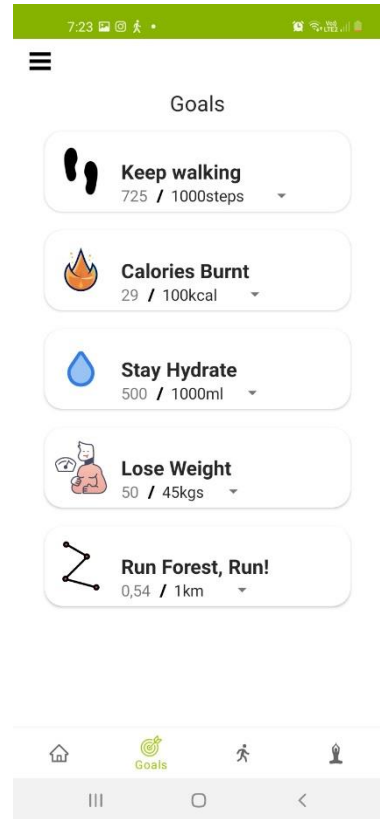
Οι διεπαφές Home, Side Menu, Goals, Training, Run, Stretch που σχεδιάστηκαν για την έκδοση που της παρούσας πτυχιακής εργασίας, παρουσιάζονται στις εικόνες 3.2-3.7:



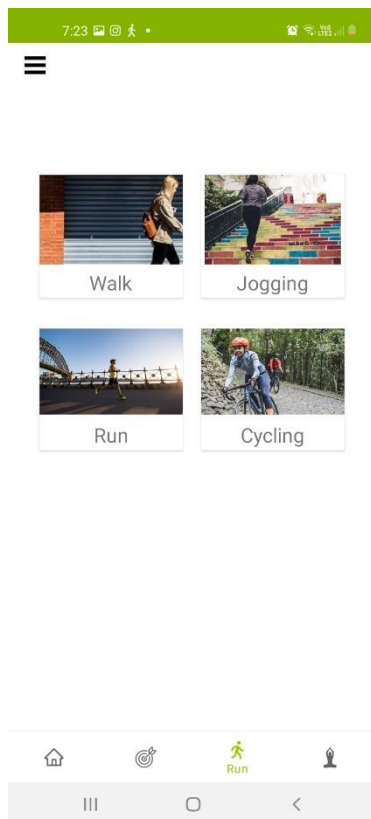
Εικόνα 3.2 Διεπαφή "Home"



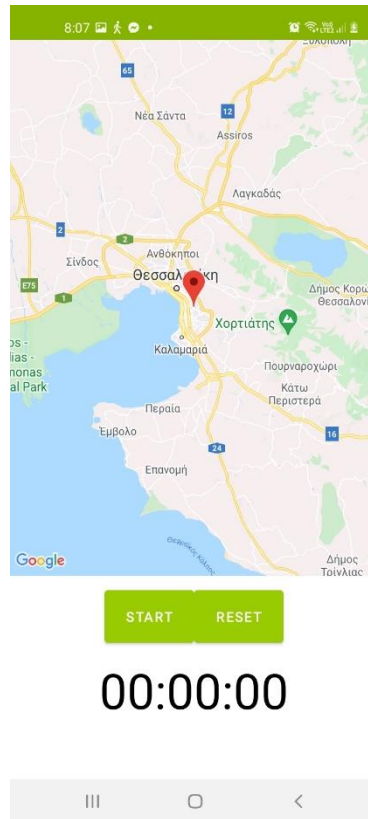
Εικόνα 3.3 "Side Menu"



Εικόνα 3.4 Διεπαφή "Goals"



Εικόνα 3.5 Διεπαφή "Training"



Εικόνα 3.6 Διεπαφή "Run"



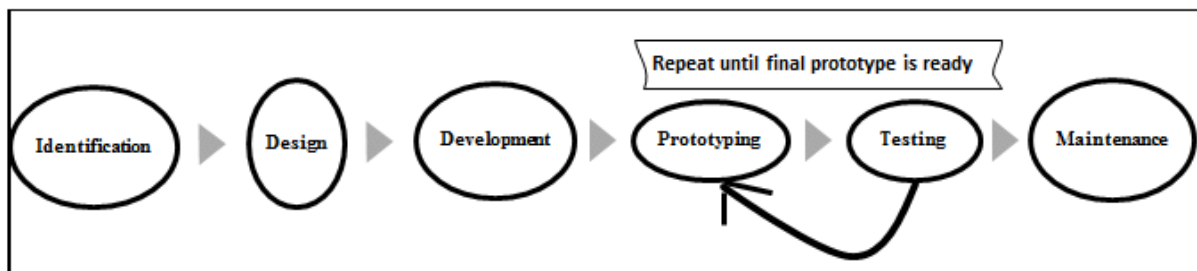
Εικόνα 3.7 Διεπαφή "Stretch"

3.2 Κύκλος ζωής ανάπτυξης εφαρμογών για κινητά (MADLC)

Ο Tejas Vithani και ο Anand Kumar το 2014 πρότειναν έναν κύκλο ζωής ανάπτυξης εφαρμογών για κινητά (MADLC) που περιλαμβάνει τις ακόλουθες φάσεις: Ταυτοποίηση, Σχεδιασμός, Ανάπτυξη, Πρωτοτυποποίηση, Δοκιμή και Συντήρηση [18].

1. Φάση ταυτοποίησης: Στην πρώτη φάση, γίνεται η συλλογή και η κατηγοριοποίηση των ιδεών, που μπορούν να προέρχονται είτε από τον πελάτη είτε από τους προγραμματιστές, είτε και από την συνεργασία τους.
2. Φάση σχεδίασης: Σε αυτήν τη φάση, η ιδέα εξελίσσεται σε έναν αρχικό σχεδιασμό της εφαρμογής (wireframing). Παρακάτω αναλύεται η διαδικασία που ακολουθήθηκε κατά την φάση σχεδίασης.
3. Φάση ανάπτυξης: Σε αυτήν τη φάση, η εφαρμογή κωδικοποιείται. Η κωδικοποίηση για διαφορετικές ενότητες του ίδιου πρωτοτύπου μπορεί να προχωρήσει παράλληλα.
4. Φάση πρωτοτυποποίησης: -Σε αυτήν τη φάση, αναλύονται οι λειτουργικές απαιτήσεις κάθε πρωτοτύπου. τα πρωτότυπα ελέγχονται και αποστέλλονται στον πελάτη για ανατροφοδότηση. Μετά την ανατροφοδότηση, οι απαιτούμενες αλλαγές υλοποιούνται μέσω της φάσης ανάπτυξης. Όταν το δεύτερο πρωτότυπο είναι έτοιμο, ενσωματώνεται με το πρώτο πρωτότυπο, δοκιμάζεται και στη συνέχεια αποστέλλεται στον πελάτη. Οι φάσεις ανάπτυξης, πρωτοτυποποίησης και δοκιμής επαναλαμβάνονται έως ότου το τελικό πρωτότυπο είναι έτοιμο.
5. Φάση δοκιμής: Ο έλεγχος των τύπων πρωτοτύπων πραγματοποιείται σε εξομοιωτή / προσομοιωτή και στη συνέχεια δοκιμές στην πραγματική συσκευή.
6. Φάση δοκιμής: Αφού ολοκληρωθεί ο έλεγχος και ληφθεί η τελευταία αξιολόγηση του πελάτη, η εφαρμογή είναι έτοιμη για δημοσίευση.
7. Φάση συντήρησης: Στην τελική φάση συλλέγονται τα σχόλια των χρηστών της εφαρμογής και γίνονται οι απαιτούμενες βελτιώσεις και διορθώνονται τυχόν σφάλματα.

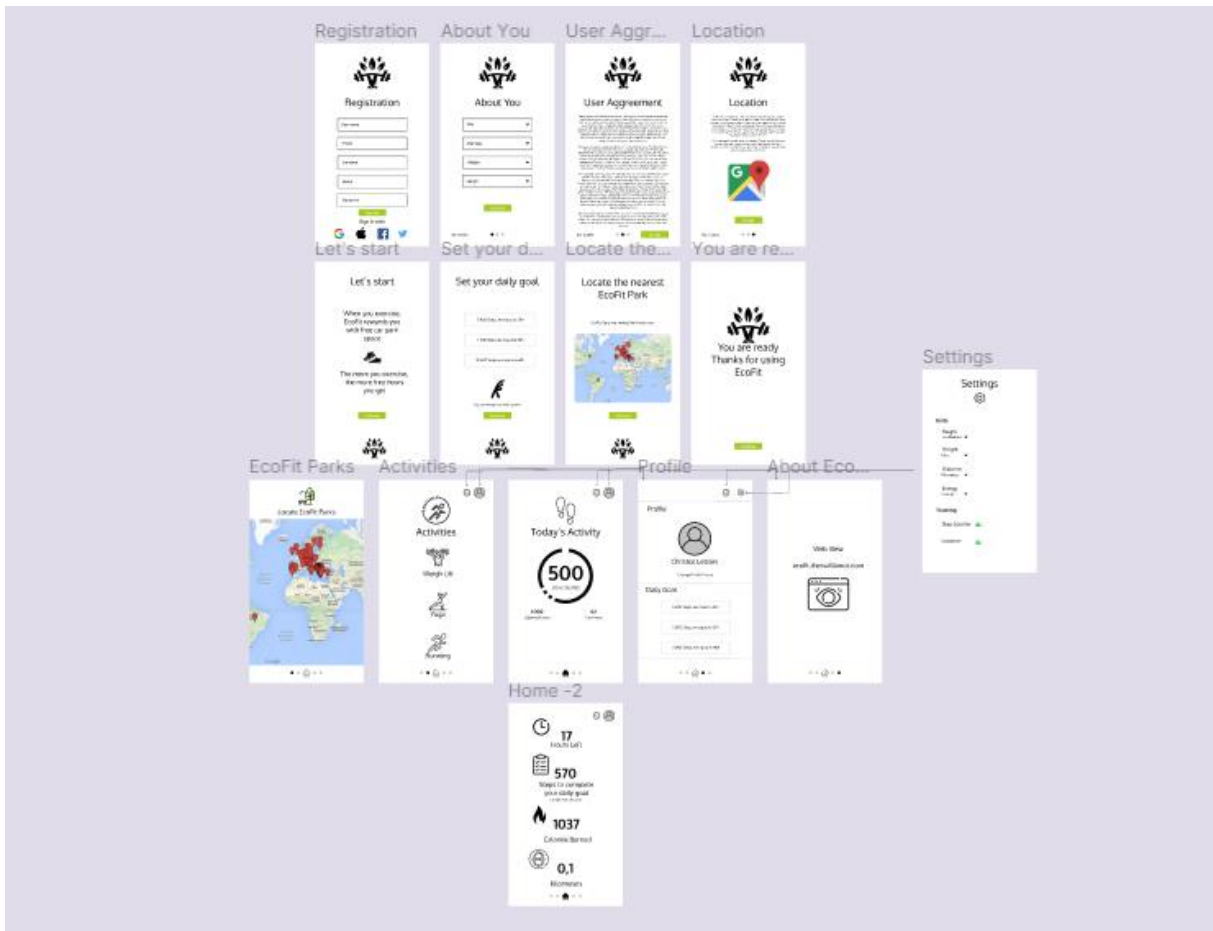
Στα σχήμα 3.1 παρουσιάζεται ο κύκλος ζωής ανάπτυξης εφαρμογών για κινητά, όπως προτάθηκε από τους Tejas Vithani και Anand Kumar [19].



Σχήμα 3.1 Mobile Application Development Lifecycle (MADLC) [19]

3.3 Φάση σχεδίασης

Κατά την φάση της σχεδίασης χρησιμοποιήθηκε η πλατφόρμα Figma για την σχεδίαση των Wireframes. Για το τελικό UI σχεδιάστηκαν αρκετά διαφορετικά wireframes. Στο σχήμα 3.2 παρουσιάζεται η πρώτη εκδοχή-σχέδιο UI. Αρχικά υπήρξε η ιδέα για έναν minimal σχεδιασμό αλλά έπειτα από την έρευνα στη θεωρία ψυχολογίας χρώματος, το συμπέρασμα ήταν πως η χρήση περισσότερων χρωμάτων και εικόνων, θα δημιουργούσαν μια πιο φιλική εφαρμογή, η οποία θα απέδιδε και θετική ενέργεια στον χρήστη. Το τελικό προϊόν UI παρουσιάζεται στην Ενότητα 3.10.



Σχήμα 3.2 Figma wireframe #1

3.4 Συλλογές Βάσης Δεδομένων

Το πρώτο βήμα έπειτα από την σχεδίαση των Wireframes ήταν η σχεδίαση και η δημιουργία της βάσης δεδομένων και των συλλογών (Collections) της, σύμφωνα με τις ανάγκες της εφαρμογής. Τα Collections αυτά είναι: Users, Metrics, Water, Records και Goals.

3.4.1 Users

Στον πίνακα Users (Εικόνα 3.8) αποθηκεύονται τα στοιχεία που εισάγει ο χρήστης στο Sign Up. Τα στοιχεία αυτά είναι:

- Όνομα,
- Επώνυμο,
- Username
- Email,
- Μετρητής επανασύνδεσης

quarafit-9b7eb + Start collection Goals Metrics Records Users > Water	Users + Add document 41Ho2wt5YsXCxAhM40Y503Ygm4x1 5tXHTLL6h7YbPM6TdNN901g3bGM2 Hg08rvwVIiY2HMHUM7Lt8aEey5b2 J6KcNpLDr03nF8V1193HoZa0Ue2 JJnkhCfVadZ2xo1VnB1fd7fJ5ro1 XT06ZzKfWg0T04zKKMEaaYJT1di1 pp3oz0kkvMSN3q0j35t54e5U77E2	4lHo2wt5YsXCxAhM40Y503Ygm4x1 + Start collection + Add field Email: "elliste12@live.com" FirstName: "Elli" Login: 1 Surname: "St" Username: "elli"
---	--	--

Εικόνα 3.8 Collection “Users”

3.4.2 Metrics

Στο collection “Metrics” αποθηκεύονται οι πληροφορίες Βάρους και Ύψους ανά χρήστη. Μπορούν να εισαχθούν αμέσως μετά την δημιουργία λογαριασμού και ή να μεταβληθούν ή να προστεθούν μετά την είσοδο στην εφαρμογή. Η δομή του Metrics παρουσιάζεται στην εικόνα 3.9.

quarafit-9b7eb + Start collection Goals Metrics > Records Users Water	Metrics + Add document 41Ho2wt5YsXCxAhM40Y503Ygm4x1 D6IkR5TA6EGyrthtMxSp	4lHo2wt5YsXCxAhM40Y503Ygm4x1 + Start collection + Add field Height: 150 Weight: 50
---	---	--

Εικόνα 3.9 Collection “Metrics”

3.4.3 Water

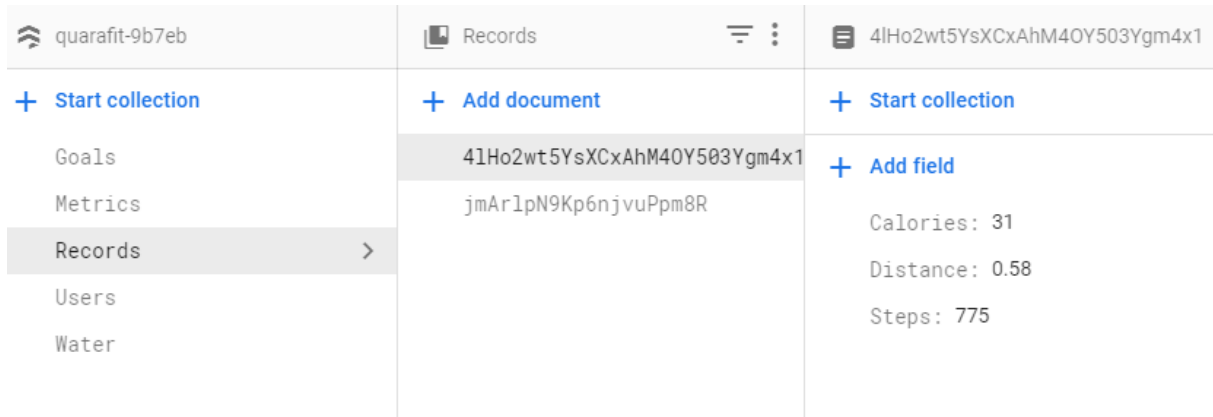
Στο Collection “Water”, που παρουσιάζεται στην εικόνα 3.10 ,αποθηκεύονται οι τιμές των ποτηριών και της ποσότητας νερού που καταναλώνει ο κάθε χρήστης ημερησίως.

quarafit-9b7eb + Start collection Goals Metrics Records Users Water >	Water + Add document 41Ho2wt5YsXCxAhM40Y503Ygm4x1	4lHo2wt5YsXCxAhM40Y503Ygm4x1 + Start collection + Add field Glass: 1 M1: 250
---	---	--

Εικόνα 3.10 Collection “Water”

3.4.4 Records

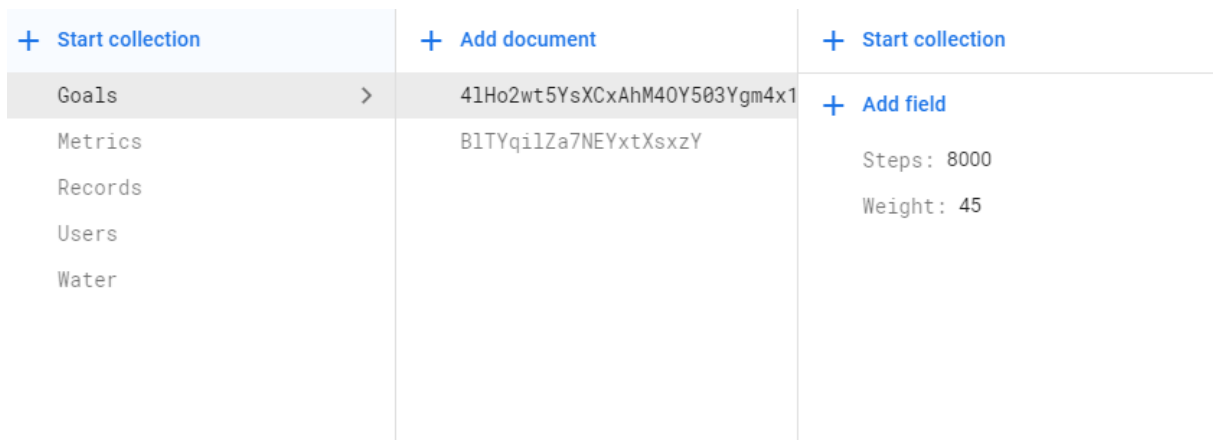
Στο Collection “Records”, που παρουσιάζεται στην εικόνα 3.11 , είναι αποθηκευμένες οι τιμές που αφορούν την καθημερινή δραστηριότητα του χρήστη. Συγκεκριμένα αποθηκεύονται οι θερμίδες που κάηκαν, τα βήματα και τα χιλιόμετρα που διανύθηκαν ημερησίως.



Εικόνα 3.11 Collection “Records”

3.4.5 Goals

Στο Collection “Goals”, που παρουσιάζεται στην εικόνα 3.12 , αποθηκεύονται οι τιμές των στόχων του κάθε χρήστη.



Εικόνα 3.12 Collection “Goals”

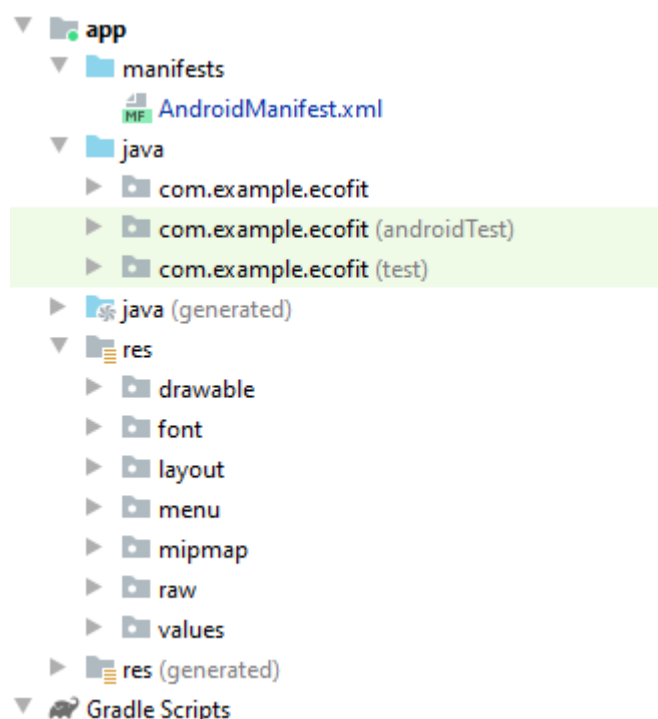
3.5 Δομή πρότζεκτ στο Android Studio

Στο σχήμα 3.3 παρουσιάζεται η δομή του πρότζεκτ της εφαρμογής στο Android studio. Το αρχείο AndroidManifest.xml περιλαμβάνει σημαντικές πληροφορίες για την εφαρμογή, για τα εργαλεία building που χρησιμοποιεί το Android, το λειτουργικό σύστημα καθώς και το Google Play. Ένα Android project μπορεί να έχει παραπάνω από ένα αρχείο manifest αν υπάρχουν περισσότερα flavors (παραλλαγές) για την εφαρμογή., όπως η ελεύθερη έκδοση μιας εφαρμογής και η πληρωμένη έκδοση της.

Ο επόμενος φάκελος του δέντρου περιλαμβάνει υποφακέλους με αρχεία java. Ο πρώτος υποφάκελος περιέχει τον πηγαίο κώδικα της java με τις μεθόδους για την λειτουργία της εφαρμογής, κι έπειτα

υπάρχουν οι υποφάκελοι `androidTest` και `test` με αρχεία `java` δοκιμής (Testing), το `ExampleInstrumentedTest` και το `ExampleUnitTest`. Στο `ExampleUnitTest.java` μπορούν να προστεθούν Unit Tests τα οποία γράφονται από τον προγραμματιστή με σκοπό να δοκιμάζουν αν λειτουργούν σωστά οι μέθοδοι του πρότζεκτ. Στο αρχείο `ExampleInstrumentedTest.java` τοποθετούνται τα Instrumented Tests που είναι tests που έχουν αναπτυχθεί για να τρέξουν σε φυσικές συσκευές [23].

Στον φάκελο `res` (resources) βρίσκονται τα αρχεία που αφορούν την σχεδίαση των διεπαφών. Τα αρχεία αυτά κατηγοριοποιούνται ανά είδος στους υποφακέλους `drawable`, `font`, `layout`, `menu`, `mipmap`, `raw`, `values`, `themes`. Ο φάκελος `layout` περιλαμβάνει τις σχεδιάσεις των διεπαφών των `activities`, `fragments` ή άλλα `views`. Στον φάκελο `drawable` υπάρχουν όλες οι εικόνες και τα σχήματα που χρησιμοποιούνται στην εφαρμογή. Στο `fonts` υπάρχουν άλλες γραμματισειρές που δεν περιέχονται στην αρχική έκδοση του Android Studio. Το αρχείο `themes` είναι αυτό που περιέχει τις μεταβλητές χρωμάτων θέματος. Στην εφαρμογή μου υπάρχει ένα θέμα για ημέρα κι ένα για νύχτα.



Σχήμα 3.3 Δομή Android πρότζεκτ

3.6 AndroidManifest.xml

Το αρχείο `AndroidManifest.xml` είναι απαιτούμενο root αρχείο για κάθε πρότζεκτ στο Android Studio. Είναι το αρχείο όπου περιγράφονται βασικές πληροφορίες για την εφαρμογή ώστε να δημιουργηθεί η εφαρμογή (build), για το λειτουργικό σύστημα του Android και το Google Play. Επίσης εδώ δηλώνεται το όνομα του πακέτου της εφαρμογής, τα διάφορα `components` της, τα δικαιώματα, αλλά και απαιτούμενα χαρακτηριστικά του λογισμικού και του υλικού (Εικόνα 3.13).

Κατά την διάρκεια του build, το package name θα αντικατασταθεί από το application ID το οποίο έχει οριστεί στο Gradle, για χρησιμοποιηθεί ως μοναδικός προσδιοριστής της εφαρμογής κατά την μεταφορά της στο Google Play. Επίσης, όσον αφορά το Google Play στο android manifest υπάρχουν χαρακτηριστικά της εφαρμογής τα οποία δείχνουν ποια κινητά είναι ικανά να χρησιμοποιήσουν την εφαρμογή από άποψη λειτουργικού συστήματος αλλά και υλικού.

Κεφάλαιο 3

Αμέσως μετά σημαντικό κομμάτι είναι τα δικαιώματα (permissions) (Εικόνα 3.14) τα οποία χρειάζεται να χρησιμοποιήσει η εφαρμογή. Τέλος ορίζονται τα διάφορα components όπως services, activities και meta-data και βιβλιοθήκες.

Για αρχή, ως προς την δομή του, είναι και αυτό σε xml και πρέπει να έχει πολύ συγκεκριμένη δομή. Το root element είναι το <manifest> για το οποίο είναι απαραίτητο να οριστεί το package attribute το οποίο περιλαμβάνει το οποίο έχει την τιμή “ com.example.ecofit”. Για αρχή αυτό χρησιμοποιείται για να οριστεί το namespace της εφαρμογής.

Το package name χρησιμεύει για να οριστεί το R class το οποίο χρησιμεύει για να έχουμε πρόσβαση στα resources του project. Η R κλάση περιέχει τους ορισμούς για όλα για όλα τα resources του project. Για παράδειγμα, στην περίπτωση μας εφόσον το package name είναι “com.example.ecofit”, τότε δημιουργείται μια κλάση com.example.ecofit.R.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ecofit">
    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="EcoFit"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.EcoFit">
        <activity android:name=".WaterActivity"></activity>
        <activity android:name=".OutTrainActivity" />
        <activity android:name=".BMICalculatorActivity" />
        <activity android:name=".ContactUsActivity" />
    </application>
</manifest>
```

Εικόνα 3.13 Δομή manifest τμήμα 1

```
<activity android:name=".BiometricsActivity" />
<activity android:name=".LaunchActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".HomeActivity" />
<activity android:name=".LoginActivity" />
<activity android:name=".SignupActivity" />

<meta-data
    android:name="com.facebook.sdk.ApplicationId"
    android:value="441275463894635" />
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyCdNgCyViLAY00qexIq_ynspsdIANkaJUI" />
<meta-data
    android:name="com.google.android.gms.version"
    android:value="12451000" />

<uses-library android:name="com.google.android.maps" />
</application>
</manifest>
```

Εικόνα 3.14 Δομή manifest τμήμα 2

Στο AndroidManifest δηλώνεται ποιο θα είναι το κύριο Activity με το οποίο θα ξεκινάει η εφαρμογή και άλλες σημαντικές πληροφορίες όπως τα meta-data και η χρήση των χαρτών. (Εικόνα 3.15)

```

<meta-data
  android:name="com.facebook.sdk.ApplicationId"
  android:value="@string/facebook_app_id" />
<meta-data
  android:name="com.google.android.geo.API_KEY"
  android:value="@string/google_maps_key" />
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />

<uses-library android:name="com.google.android.maps" />
</application>

```

Εικόνα 3.15 meta-data

3.6.1 Δικαιώματα

Τα δικαιώματα τα οποία ορίζονται στο manifest.xml έχουν ιδιαίτερη σημασία καθώς επηρεάζουν τις λειτουργίες που μπορεί να περιλαμβάνει η εφαρμογή. Η δήλωση του permission γίνεται μέσα στο root element, δηλαδή το manifest και είναι της μορφής “<uses-permission android:name=“PERMISSION_NAME”/>”.

Τα δικαιώματα τα οποία χρησιμοποιούνται στην εφαρμογή είναι το ACCESS_FINE_LOCATION, το ACCESS_BACKGROUND_LOCATION, το ACCESS_COARSE_LOCATION, το FOREGROUND_SERVICE και το INTERNET (Εικόνα 3.16).

```

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.INTERNET" />

```

Εικόνα 3.16 Permissions

Το ACCESS_FINE_LOCATION και το δικαίωμα ACCESS_COARSE_LOCATION παραχωρούνται για να μπορεί η εφαρμογή να έχει πρόσβαση στη τοποθεσία του χρήστη. Επιτρέπει στο API να προσδιορίσει όσο το δυνατόν ακριβέστερη τοποθεσία από τους διαθέσιμους παρόχους τοποθεσίας, όπως το Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS), καθώς και τα δεδομένα WiFi και κινητής τηλεφωνίας. [23]

Το δικαίωμα INTERNET επιτρέπει την πρόσβαση της εφαρμογής στο διαδίκτυο. Με αυτό το δικαίωμα η εφαρμογή, θα μπορεί να αλληλεπιδράσει με το web service και να έχει πρόσβαση στα βασικά δεδομένα της εφαρμογής. Χωρίς αυτό το δικαίωμα, η εφαρμογή δεν μπορεί να λειτουργήσει.

3.7 Βιβλιοθήκες

3.7.1 GOOGLE PLAY SERVICES

Η βιβλιοθήκη Google Play Services παρέχει διεπαφές για την επικοινωνία με πολλές σημαντικές υπηρεσίες της Google. Χρησιμοποιώντας τη βιβλιοθήκη είναι δυνατόν οι χρήστες της εφαρμογής να πάρουν πρόσβαση στις υπηρεσίες της Google χρησιμοποιώντας τα διαπιστευτήριά τους. Σημαντικό στοιχείο της βιβλιοθήκης επίσης είναι ότι έχει πολύ μικρό αντίκτυπο στη συνολική μνήμη της εφαρμογής. Συγκεκριμένα στην παρούσα εφαρμογή παρέχονται οι υπηρεσίες (services) : maps και location για να την εμφάνιση του Google Map και την εύρεση της ακριβής τοποθεσίας του χρήστη.

```
implementation 'com.google.android.gms:play-services-maps:17.0.0'
```

```
implementation 'com.google.android.gms:play-services-location:18.0.0'
```

3.7.2 AndroidX test

Στην εφαρμογή έχει προστεθεί η βιβλιοθήκη AndroidX ώστε νέες λειτουργίες που έχουν προστεθεί σε νέες εκδόσεις εφαρμογών, να τρέχουν σωστά και σε προηγούμενες εκδόσεις της εφαρμογής. Έχει αντικαταστήσει την Support Library. Στο αρχείο build.gradle μέσα στο dependencies πρέπει να προστεθούν οι παρακάτω γραμμές:

```
androidTestImplementation 'androidx.test.ext:junit:1.1.1'
```

```
androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
```

Ουσιαστικά προσφέρει συμβατότητα προς τα πίσω για νεότερα API, βοηθητικές κλάσσεις και αποσφαλμάτωση,

3.7.3 Facebook Login for Android

Το Facebook SDK για Android επιτρέπει στους χρήστες να συνδέονται στις εφαρμογές με τον Facebook λογαριασμό. Όταν οι χρήστες συνδέονται στην εφαρμογή μέσω Facebook, μπορούν να εκχωρήσουν δικαιώματα στην εφαρμογή, για την ανάκτηση πληροφοριών (πχ Όνομα, email) ή για την εκτέλεση ενεργειών στο Facebook για λογαριασμό τους (πχ Δημοσίευση ημερήσιων βημάτων).

```
implementation 'com.facebook.android:facebook-android-sdk' [9]
```

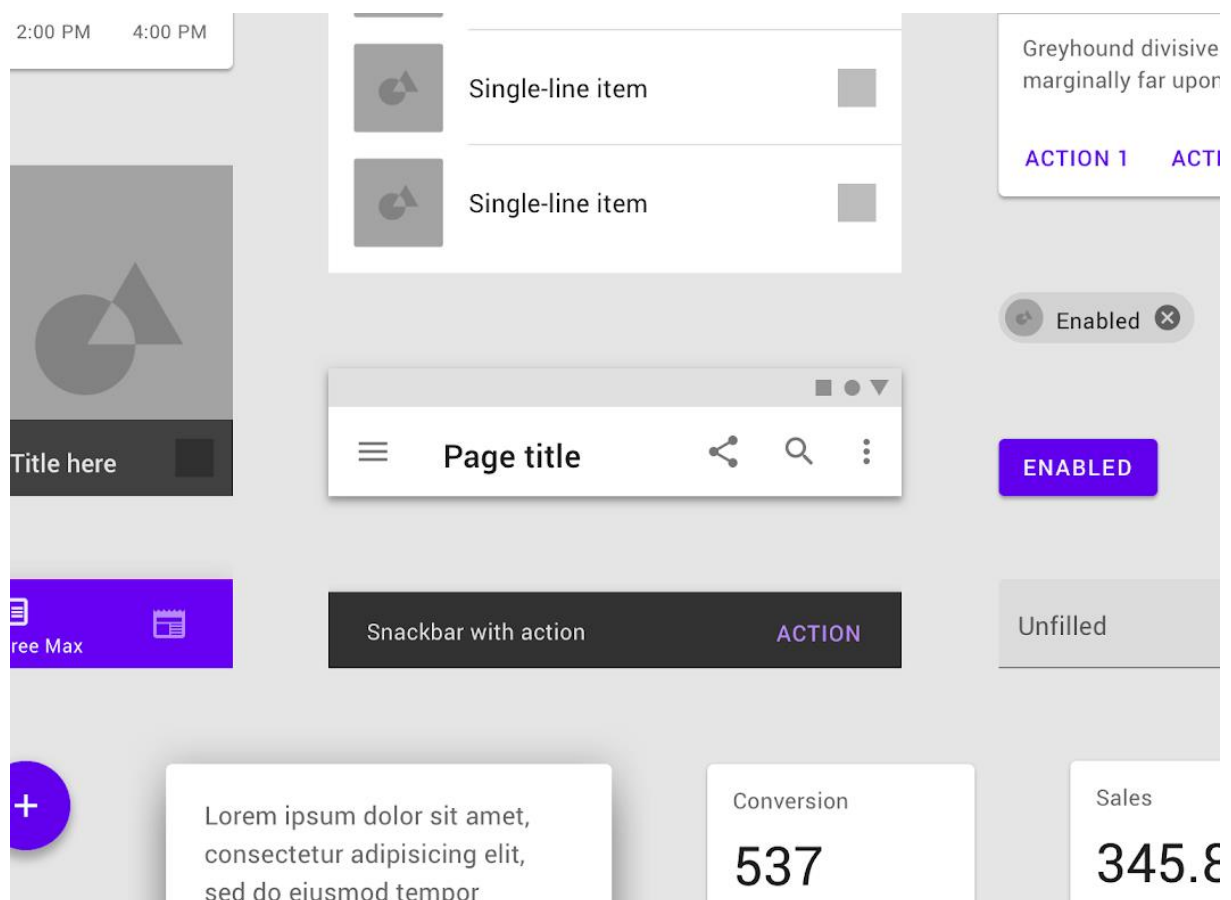
3.7.4 Material Design

Το Material [20] είναι ένα σύστημα σχεδίασης open source που δημιουργήθηκε από την Google για να βοηθήσει ομάδες να δημιουργήσουν υψηλής ποιότητας ψηφιακές εμπειρίες για Android, iOS[35], Flutter [36] και τον ιστό. Στοχεύει στην υποστήριξη πολλαπλών περιβαλλόντων διεπαφών. Ένας άλλος κύριος στόχος είναι η μίμηση πραγματικών αντικειμένων ώστε να βελτιστοποιηθεί η εμπειρία των χρηστών. Το material design ξεκίνησε στα μέσα του 2014 με σκοπό τη συνέπεια σε όλα τα περιβάλλοντα.

Τα τρία κύρια χαρακτηριστικά του material design είναι:

1. Η μεταφορά του υλικού.
Μεταφορά της έννοιας του υλικού και της αφής και η προσφορά χαρακτηριστικών των οποίων οι χρήστες βρίσκουν γνώριμα. Χαρακτηριστικά που καταλαβαίνει κατευθείαν ο χρήστης, με τα οποία όταν αλληλεπιδρά έχει μια πολύ καλή ‘μαντεψιά’ για το τι θα ακολουθήσει.
2. Έντονα και διαλεγμένα με μεγάλη προσοχή στοιχεία, τα οποία είναι ευχάριστα στο μάτι.
Ο σχεδιασμός είναι έντονος και προσεγμένος ώστε οι χρήστες να μπορούν να εστιάζουν στα κατάλληλα σημεία και να καταλαβαίνουν γρήγορα το μήνυμα. Επίσης, γίνεται προσεγμένη χρήση των χρωμάτων και των κενών ώστε να αξιοποιείται η επιφάνεια με τον καλύτερο δυνατό τρόπο.
3. Η αλληλεπίδραση με τον χρήστη γίνεται μέσω κινήσεων.
Το αποτέλεσμα της κάθε κίνησης γίνεται αντιληπτό εύκολα και είναι εμφανές. Επίσης, τα αποτελέσματα είναι απλά και ‘καθαρά’ χωρίς επιπλέον αχρείαστα χαρακτηριστικά.

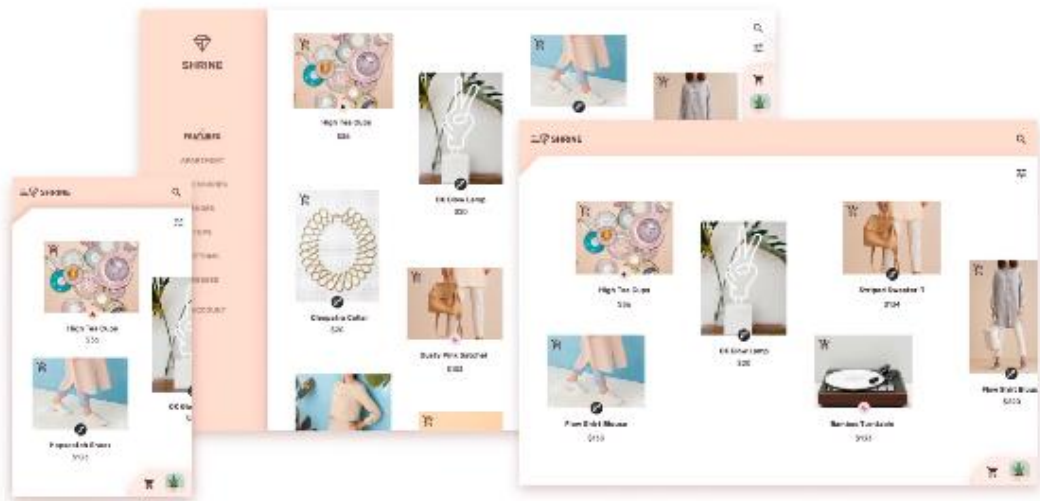
Τα συστατικά στοιχεία του Material είναι διαδραστικά δομικά στοιχεία για τη δημιουργία διεπαφής χρήστη και περιλαμβάνουν ένα ενσωματωμένο σύστημα καταστάσεων για την επικοινωνία των καταστάσεων: εστίασης, επιλογής, ενεργοποίησης, απενεργοποίησης, μεταφοράς, σφάλματος, drag, press (Σχήμα 3.4).



Σχήμα 3.4 Material Components [20]

Αυτά τα Components καλύπτουν ένα εύρος αναγκών διεπαφής, όπως:

- Οθόνη-Display: Τοποθέτηση και οργάνωση περιεχομένου χρησιμοποιώντας στοιχεία όπως κάρτες, λίστες και φύλλα.
- Πλοήγηση-Navigation: Επιτρέποντας στους χρήστες να μετακινούνται μέσω του προϊόντος χρησιμοποιώντας στοιχεία όπως navigation drawers (“συρτάρια πλοήγησης”) και καρτέλες (Σχήμα 3.5).
- Ενέργειες-Actions: Επιτρέποντας στους χρήστες να εκτελούν εργασίες χρησιμοποιώντας στοιχεία όπως το floating action button.
- Είσοδος-Input: Επιτρέποντας στους χρήστες να εισάγουν πληροφορίες ή να κάνουν επιλογές χρησιμοποιώντας στοιχεία όπως πεδία κειμένου, μάρκες και στοιχεία ελέγχου επιλογής.
- Επικοινωνία-Communication: Προειδοποίηση των χρηστών για βασικές πληροφορίες και μηνύματα χρησιμοποιώντας στοιχεία όπως snackbars, banners και dialogs.



Σχήμα 3.5 Navigation drawers και tabs [20]

3.7.5 Animation

Για να κάνω πιο παραστατική τη δραστηριότητα “Water”, κατέβασα ένα gif το οποίο απεικονίζει ένα ποτήρι νερό, ώστε κάθε φορά που ο χρήστης κάνει κλικ (για να πει νερό) το ποτήρι αδειάζει (Εικόνες 3.17-3.20) Χρησιμοποιήθηκαν οι βιβλιοθήκες Lottie και GIFLib.

```
<com.airbnb.lottie.LottieAnimationView
    android:layout_marginTop="150dp"
    android:id="@+id/animationView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:lottie_rawRes="@raw/wateranimation"
    app:lottie_autoPlay="false"
    app:lottie_loop="false"/>
```

Εικόνα 3.17 XML κώδικας με το Lottie Animation

3.7.5.1 Lottie

Το Lottie είναι μια βιβλιοθήκη που μπορεί να χρησιμοποιηθεί για κινητά Android αλλά και για iOS. Αναλύει τα κινούμενα σχέδια του Adobe After Effects[29] που εξάγονται ως json με το extension Bodymovin [38], το οποίο μετατρέπει τα animations σε html, και τα καθιστά εγγενή στο κινητό [21].

```
def lottieVersion = "3.4.0"
```

```
implementation "com.airbnb.android:lottie:$lottieVersion"
```

3.7.5.2 Διαχείριση καρτέ των gifs

Για να χρησιμοποιηθούν να διαχειριστούν τα καρτέ της βιβλιοθήκης Lottie προστέθηκε η βιβλιοθήκη android-gif-drawable [22]. Το πακέτο GIFLib μέσω JNI χρησιμοποιείται για την απόδοση καρτέ. Η μέθοδοι που παρέχονται για την διαχείριση των καρτέ είναι οι εξής:

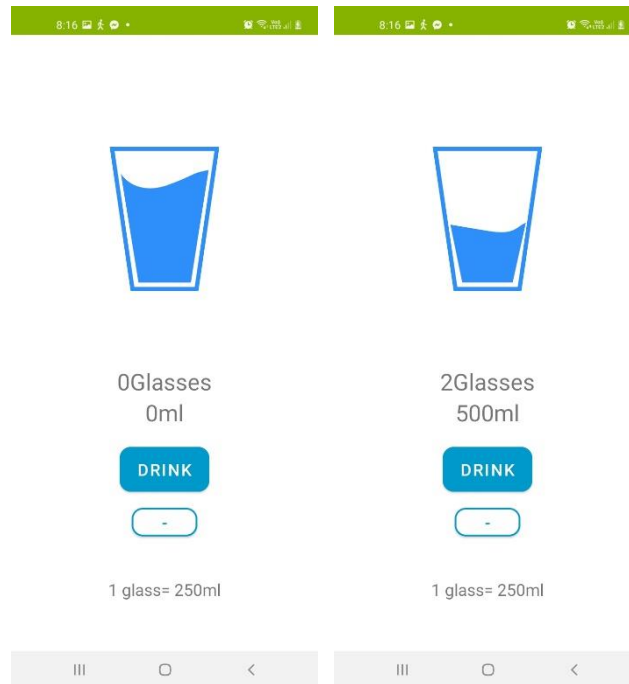
- `stop()` - stops the animation, can be called from any thread
- `start()` - starts the animation, can be called from any thread
- `isRunning()` - returns whether animation is currently running or not
- `reset()` - rewinds the animation, does not restart stopped one
- `setSpeed(float factor)` - sets new animation speed factor, eg. passing 2.0f will double the animation speed
- `seekTo(int position)` - seeks animation (within current loop) to given `position` (in milliseconds)
- `getDuration()` - returns duration of one loop of the animation
- `getCurrentPosition()` - returns elapsed time from the beginning of a current loop of animation

Εικόνα 3.18 Μέθοδοι GIFLib [22]

Στην εφαρμογή χρησιμοποίησα την `playAnimation()` μέθοδο για να ξεκινήσει να αδειάζει το ποτήρι (σχήματα 3.5,3.6), όταν ο χρήστης κάνει κλικ στο να αυξήσει τα ποτήρια που ήπια.

```
glassCounter.setText(glassCountInt+" Glasses");
mlCounter.setText(mlCountInt+" ml");
waterAnimation.playAnimation();
```

Εικόνα 3.19 Java μέθοδος `playAnimation()`



Εικόνα 3.20 Ενεργοποίηση playAnimation() (γραφικά)

Circle images

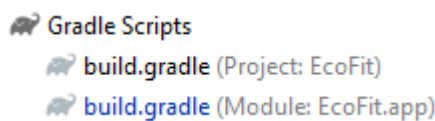
Για το στρόγγυλο σχήμα στις φωτογραφίες του χρήστη χρησιμοποιείται η βιβλιοθήκη: 'de.hdodenhof:circleimageview:3.1.0' [35].

3.8 Gradle

3.8.1 Γενικά

Το Gradle είναι open source λογισμικό το οποίο χρησιμοποιείται για την αυτοματοποίηση του Build μιας εφαρμογής και προσφέρει ευελιξία καθώς με τη χρήση του επιτρέπεται η δημιουργία Build από διάφορα είδη λογισμικού. Η εργαλειοθήκη του Gradle στο Android Studio δίνει τη δυνατότητα στον προγραμματιστή να επέμβει στη διαδικασία του Build και να κάνει παραμετροποιήσεις σύμφωνα με τις ανάγκες του πρότζεκτ του. Σε ένα πρότζεκτ μπορούν να υπάρξουν παραπάνω από ένα build και το καθένα να παρέχει διαφορετικές λειτουργίες. Οι παραλλαγές ονομάζονται flavors και μπορεί για παράδειγμα να υπάρχει ένα free demo flavor της εφαρμογής με περιορισμένες υπηρεσίες, κι ένα flavor επί πληρωμή με ολοκληρωμένη λειτουργικότητα.

Σε κάθε πρότζεκτ όπως φαίνεται στο Σχήμα 3.6 υπάρχουν δύο build.gradle scripts, το Top-level (Project) και το Module-level (:app). Η ανάπτυξη τη εφαρμογής έγινε στο Android Studio Version 4. Στις εκδόσεις 3.5 και παλαιότερα, η δομή του φακέλου και τα αρχεία Gradle Scripts ήταν διαφορετικά.



Σχήμα 3.6 Gradle Scripts

Στο Top-level (Project) καθορίζονται οι ρυθμίσεις του Build που ισχύουν σε όλα τα modules και τα repositories και τα dependencies που είναι κοινά στο πρότζεκτ.

Στο Module-level (:app) δηλώνονται και περιέχονται βασικές πληροφορίες όπως η ελάχιστη απαιτούμενη έκδοση συστήματος Android, το applicationId, η έκδοση της εφαρμογής, το compile Sdk Version και τα dependencies. Τα dependencies που χρησιμοποιούνται στην εφαρμογή φαίνονται στο σχήμα 3.7 .

```

apply plugin: 'com.google.gms.google-services'

dependencies {
    //facebook
    implementation 'com.facebook.android:facebook-android-sdk:[5,6]'
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:26.5.0')
    // Declare the dependency for the Firebase SDK for Google Analytics
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'com.google.android.material:material:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.firebase:firebase-auth:19.2.0'
    implementation 'com.google.firebase:firebase-firestore:22.1.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'com.google.android.gms:play-services-auth:19.0.0'
    testImplementation 'junit:junit:4.+
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'androidx.annotation:annotation:1.1.0'

    //maps location
    implementation 'com.google.android.gms:play-services-maps:17.0.0'
    implementation 'com.google.android.gms:play-services-location:18.0.0'

    //gif
    implementation 'pl.droidsonroids.gif:android-gif-drawable:1.2.19'
    def lottieVersion = "3.4.0"
    implementation "com.airbnb.android:lottie:$lottieVersion"

```

Σχήμα 3.7 Dependencies

3.9 Activities

3.9.1 Launch Activity

Το Launch Activity είναι το Activity που έχει δηλωθεί στο Manifest ως το Activity με το οποίο θα ξεκινάει η εφαρμογή (Εικόνα 3.21). Η γραφική διεπαφή παρουσιάζεται στην Εικόνα 3.22.

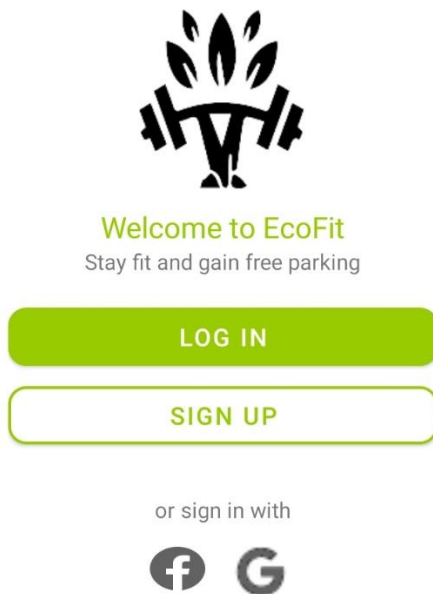
```

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

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

```

Εικόνα 3.21 Main Launcher



Εικόνα 3.22 Διεπαφή Launch Activity

Για την σύνδεση στην εφαρμογή με τον λογαριασμό του Facebook ή και του Google Account χωρίς την δημιουργία άλλου λογαριασμού, δημιούργησα facebook και google buttons (Εικόνα 3.23). Το facebook παρέχει μέθοδο κι έτοιμο γραφικό κουμπί για να συνδεθείς μέσω αυτού. Για να αλλάξω τον σχεδιασμό καθώς δεν ήθελα να φαίνεται το προσχεδιασμένο «Σύνδεση με τον λογαριασμό», έθεσα το κείμενο του κουμπιού ως κενό (setText(“”)), κι επίσης έθεσε τα Bounds ως null για να υπάρχει χώρος μόνο για το “f”. Έπειτα με τις μεθόδους που παρέχονται από τις βιβλιοθήκες google και facebook που έχω κάνει import, επιτρέπεται η απευθείας σύνδεση χρησιμοποιώντας τον αντίστοιχο λογαριασμό (Εικόνα 3.24).

```
facebookBtn =findViewById(R.id.facebookBtn);
facebookBtn.setBackgroundResource(R.drawable.facebook);
facebookBtn.setText("");
facebookBtn.setCompoundDrawablesWithIntrinsicBounds( left: null, top: null, right: null, bottom: null);

googleAccountLogin=findViewById(R.id.googleAccBtn);
login=findViewById(R.id.logInBtn1);
signup=findViewById(R.id.signUpBtn1);
firebaseAuth = FirebaseAuth.getInstance();
```

Εικόνα 3.23 Αρχικοποίηση Facebook & Google buttons

```

callbackManager = CallbackManager.Factory.create();
facebookBtn.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        Log.d(TAG, msg: "onSuccess"+loginResult);
        handleFacebookToken(loginResult.getAccessToken());
    }
    @Override
    public void onCancel() {
        Log.d(TAG, msg: "onCancel");
    }
    @Override
    public void onError(FacebookException error) {
        Log.d(TAG, msg: "onError"+error);
    }
});

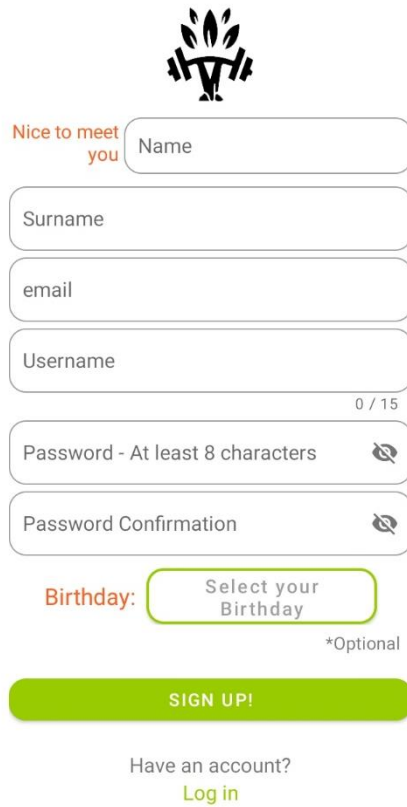
//google account login
creatGoogleAccountRequest();
findViewById(R.id.googleImg).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { signIn(); }
});

```

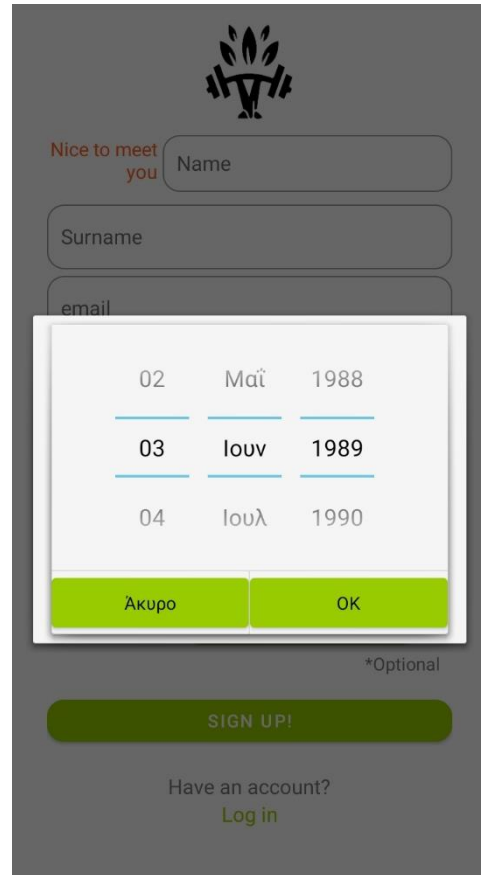
Εικόνα 3.24 Μέθοδοι σύνδεσης μέσω Facebook & Google

3.9.2 Sign Up Activity

Όταν ο χρήστης επιλέξει το Sign Up στην αρχική διεπαφή, εμφανίζεται η φόρμα καταχώρησης βασικών προσωπικών δεδομένων- στοιχείων για τη δημιουργία λογαριασμού (Εικόνα 3.25). Για την εισαγωγή της ημερομηνίας γέννησης χρησιμοποιήθηκε το Calendar, το οποίο καλείται από το Button “Select your Birthday” (Εικόνα 3.26). Θεωρώντας ότι εν έτει 2021, κάποιον άτομο λιγότερο από 12 ετών είναι μικρός για να χρησιμοποιήσει κινητό, έθεσα ως Calendar Year amount=12 (Εικόνα 3.27), έτσι ώστε να μπορούν να καταχωρηθούν γεννηθέντες του 2009 και νωρίτερα. Με το πάτημα του κουμπιού καλείται η μέθοδος onClick() η οποία με τη σειρά της δημιουργεί αντικείμενο DatePickerDialog για να εμφανίσει το ημερολόγιο το οποίο μέσω του setListener και της onDateSet, αποθηκεύει την επιλεγμένη ημερομηνία.



Εικόνα 3.25 Διεπαφή Sign Up Activity



Εικόνα 3.26 Birthday Calendar

```

birthCalendar=findViewById(R.id.birthCalendar);
final Calendar calendar= Calendar.getInstance();
calendar.add(Calendar.YEAR, amount: -12);
final int year=calendar.get(Calendar.YEAR);
final int month=calendar.get(Calendar.MONTH);
final int day=calendar.get(Calendar.DAY_OF_MONTH);

//birthday
birthCalendar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DatePickerDialog datePickerDialog=new DatePickerDialog(
            context: SignupActivity.this, android.R.style.Theme_Holo_Light_Dialog_MinWidth, setListener, year, month, day);
        datePickerDialog.getDatePicker().setMaxDate(calendar.getTimeInMillis());
        datePickerDialog.show();
    }
});

setListener=new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
        month=month;
        String date=day+"/"+month+"/"+year;
        birthCalendar.setText(date);
    }
};
    
```

Εικόνα 3.27 Calendar Αρχικοποίηση και DatePicker

3.9.3 BMI Calculator Activity

Στο BMI Calculator Activity ο χρήστης αν θέλει εισάγει το βάρος του σε κιλά και το ύψος του σε εκατοστά, για να υπολογιστεί ο Δείκτης Μάζας Σώματος (BMI) (Εικόνα 3.28). Ο υπολογισμός γίνεται με βάση την εξίσωση: $BMI = \text{βάρος} / (\text{ύψος} * \text{ύψος}) [2]$. Έπειτα ο χρήστης βλέπει κατά πόσο το βάρος του είναι φυσιολογικό, ανάλογα με τη τιμή του BMI (Εικόνα 3.29). Ο υπολογισμός BMI μπορεί να γίνει και κάποια άλλη χρονική στιγμή από το Fragment των ατομικών του στοιχείων, την επιλογή του Προφίλ στο Drawer Shift- Side Menu. Η εύρεση BMI δίνει τη δυνατότητα να ελέγξει αν βρίσκεται στα κιλά που θα πρέπει να είναι για να είναι-μείνει υγιής.

```
String heightS=height.getText().getText().toString();
float weightValue=Float.parseFloat(weightS);
float heightValue=Float.parseFloat(heightS)/100;

userID = firebaseAuth.getCurrentUser().getUid();

DocumentReference documentReference = db.collection( collectionPath: "Metrics").document(userID);
documentReference.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
    //on success
    @Override
    public void onComplete(@NonNull Task<DocumentSnapshot> task) {
        DocumentSnapshot document = task.getResult();
        Map<String, Object> metrics = new HashMap<>();
        metrics.put("Weight", weightValue);
        metrics.put("Height", heightValue);
        documentReference.set(metrics);
    }
    //on failure
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.w( tag: "ERROR", msg: "Error getting documents.", e);
    }
});

result.setText(BMIresultCalc(weightValue,heightValue));
```

Εικόνα 3.28 Ενημέρωση Βάσης και εμφάνιση BMI αποτελέσματος

```
private String BMIresultCalc(float weight, float height){
    String BMIResult;
    float bmi;

    bmi=weight/(height*height);

    if(bmi<16)
        BMIResult="Severely Under weight";
    else if(bmi<18.5)
        BMIResult="Under weight";
    else if(bmi>=18.5 && bmi<=24.5)
        BMIResult="Normal weight";
    else if(bmi>=25 && bmi<=29.9)
        BMIResult="Overweight";
    else
        BMIResult="Obese";

    calculation =bmi+"\n"+BMIResult;
    return calculation;
}
```

Εικόνα 3.29 Υπολογισμός BMI

3.9.4 Home Activity

Το Home Activity είναι το Activity με το οποίο θα εισέρχεται στην εφαρμογή αφού πρώτα έχει γίνει σύνδεση με τα διαπιστευτήρια. Στο Home Activity «φορτώνονται» τα Menu “bottom_menu” (Εικόνα 3.300) και “side_menu”. Ουσιαστικά είναι το κεντρικό Activity και σε αυτό φορτώνονται τα Fragments Home, Goals, Out Train και τέλος το Stretch. Αυτά τα Fragments εμφανίζονται ανάλογα με την επιλογή του χρήστη στο Bottom Menu. Για να εμφανίζεται το Fragment που έχει επιλέξει ο χρήστης στο bottom menu, καλείται η μέθοδος στο σχήμα η οποία με τον έλεγχο switch, θέτει στο container τη τιμή του επιλεγμένου Fragment (Εικόνα 3.31). Το side_menu εμφανίζεται χρησιμοποιώντας την μέθοδο navigationDrawer() (Εικόνα 3.32). Έπειτα όπως και για το bottom menu, καλείται η αντίστοιχη μέθοδος για να «διαβάσει» την επιλογή του χρήστη από το side menu (Εικόνα 3.33). Και εδώ με τον αντίστοιχο έλεγχο switch εμφανίζεται το κατάλληλο Fragment ή Activity. Στο side menu ο χρήστης έχει και την επιλογή να αποσυνδεθεί από την εφαρμογή. Αυτό επιτυγχάνεται με τη μέθοδο **firebaseAuth.signOut()** (Εικόνα 3.33).

```
bottomNavigationView=findViewById(R.id.bottomNav);
bottomNavigationView.setOnNavigationItemSelectedListener(bottomNavMethod);
getSupportFragmentManager().beginTransaction().replace(R.id.container,new HomeFragment()).commit();
```

Εικόνα 3.30 Χτίσιμο bottom_menu

```
private BottomNavigationView.OnNavigationItemSelectedListener bottomNavMethod= new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        Fragment selectedFragment=null;

        switch (item.getItemId()){
            case R.id.nav_home:
                selectedFragment=new HomeFragment();
                break;
            case R.id.nav_goal:
                selectedFragment=new GoalsFragment();
                break;
            case R.id.nav_run:
                selectedFragment=new OutTrainFragment();
                break;
            case R.id.nav_stretch:
                selectedFragment=new StretchFragment();
                break;
        }

        getSupportFragmentManager().beginTransaction().replace(R.id.container,selectedFragment).commit();

        return true;
    }
};
```

Εικόνα 3.31 Μέθοδος επιλογής αντικειμένου στο Bottom Menu

```
private void navigationDrawer() {
    navigationView.bringToFront();
    navigationView.setNavigationItemSelectedListener(this);
    navigationView.setCheckedItem(R.id.nav_home);

    menuIcon.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (drawerLayout.isDrawerVisible(GravityCompat.START))
                drawerLayout.closeDrawer(GravityCompat.START);
            else drawerLayout.openDrawer(GravityCompat.START);
        }
    });
}
```

Εικόνα 3.32 Μέθοδος navigationDrawer()

```
public boolean onNavigationItemSelectedListener(@NonNull MenuItem item) {
    int id = item.getItemId();
    Fragment selectedFragment=null;
    switch (id){
        case R.id.nav_home:
            selectedFragment=new HomeFragment();
            break;
        case R.id.nav_profile:
            selectedFragment=new ProfileFragment();
            break;
        case R.id.nav_settings:
            selectedFragment=new SettingsFragment();
            break;
        case R.id.nav_contact:
            Intent intentContact = new Intent( packageContext: HomeActivity.this, ContactUsActivity.class);
            startActivity(intentContact);
            finish();
            break;
        case R.id.nav_logout:
            firebaseAuth.signOut();
            Intent intent = new Intent( packageContext: HomeActivity.this, LaunchActivity.class);
            startActivity(intent);
            finish();
            break;
    }
    drawerLayout.closeDrawer(GravityCompat.START);
    if (selectedFragment!=null) {
        getSupportFragmentManager().beginTransaction().replace(R.id.container, selectedFragment).commit();
    }
    return true;
}
```

Εικόνα 3.33 Μέθοδοι item selected and logout

3.9.5 Water Activity

Στο Water Activity μεταβαίνει ο χρήστης από το Home Fragment για να προσθέσει ή να αφαιρέσει τα την ποσότητα νερού που κατανάλωσε ημερησίως. Αρχικά καλείται η μέθοδος updateGlasses() η οποία ενημερώνει την εφαρμογή με στοιχεία από την βάση, αν έχουν καταχωρηθεί προηγουμένως (Εικόνα 3.34). Εδώ όπως προανέφερα χρησιμοποιήσα animation για να αδειάζει το ποτήρι νερού όταν ο χρήστης

Κεφάλαιο 3

κάνει κλικ στο κουμπί «+» και ενημερώνεται ο πίνακας Water. Αν πατηθεί το κουμπί «-», μειώνονται τα ml που έχει καταναλώσει και ενημερώνεται αντίστοιχα ο πίνακας. (Εικόνα 3.35)

```
drinkBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        TextView glasses,ml;
        //fetch db
        userID = firebaseAuth.getCurrentUser().getUid();

        DocumentReference documentReference = db.collection( collectionPath: "Water").document(userID);
        documentReference.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {...}
            @Override
            public void onFailure(@NonNull Exception e) {...}
        });

        glassCountInt++;
        mlCountInt+=250;

        Map<String,Object> water = new HashMap<>();
        water.put ("Glass",glassCountInt);
        water.put ("Ml",mlCountInt);
        documentReference.set(water);

        glassCounter.setText(glassCountInt+" Glasses");
        mlCounter.setText(mlCountInt+" ml");
        waterAnimation.playAnimation();
    }
});

decGlassBtn.setOnClickListener(new View.OnClickListener() {...});
```

Εικόνα 3.34 Εισαγωγή δεδομένων στο Collection Water

```
public void updateGlasses(){

    userID = firebaseAuth.getCurrentUser().getUid();
    final FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();

    //fetch data from db

    DocumentReference documentReference = db.collection( collectionPath: "Water").document(userID);
    documentReference.addSnapshotListener( activity: this, new EventListener<DocumentSnapshot>() {
        @Override
        public void onEvent(@Nullable DocumentSnapshot value, @Nullable FirebaseFirestoreException error) {
            if (value != null){

                glassCounter.setText(String.valueOf(value.get("Glass")));
                mlCounter.setText(String.valueOf(value.get("Ml")));
            }
        }
    });
}
```

Εικόνα 3.35 Ενημέρωση δεδομένων εφαρμογής από την βάση δεδομένων

3.9.6 Contact Us

Το Activity Contact Us καλείται από το Side Menu. Μέσω αυτού του activity μπορεί ο χρήστης να στείλει ηλεκτρονικό μήνυμα στην διεύθυνση του ηλεκτρονικού ταχυδρομείου που έχουμε δηλώσει (Εικόνα 3.36), για να επικοινωνήσει μαζί μας. Αφού συμπληρώσει τα πεδία Θέμα και Σώμα, μεταβαίνει στον Email Client που έχει επιλέξει για να αποστείλει το μήνυμά του.

```

sendBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String subject=emailSubject.getEditText().getText().toString().trim();
        String body=emailBody.getEditText().getText().toString().trim();
        String emailAddress="info@thessalliance.com";

        if(subject.isEmpty()){
            Toast.makeText( context: ContactUsActivity.this, text: "Please add Subject",Toast.LENGTH_SHORT).show();
        }else if(body.isEmpty()){
            Toast.makeText( context: ContactUsActivity.this, text: "Please add a Message",Toast.LENGTH_SHORT).show();
        }else {
            String mail="mailto:"+emailAddress +
                "?&subject="+ Uri.encode(subject)+
                "&body="+ Uri.encode(body);
            Intent intent=new Intent(Intent.ACTION_SENDTO);
            intent.setData(Uri.parse(mail));

            try {
                startActivity(Intent.createChooser(intent, title: "Send Email.."));
            }catch (Exception e){
                Toast.makeText( context: ContactUsActivity.this, text: "Exception: "+e.getMessage(),Toast.LENGTH_SHORT).show();
            }
        }
    }
});

```

Εικόνα 3.36 Μέθοδος αποστολής ηλεκτρονικών μηνυμάτων

3.10 Fragments

3.10.1 Home Fragment

Το Fragment που έχει επιλεγεί να εμφανίζεται κατά την είσοδο στην εφαρμογή, είναι το Home Fragment στο οποίο υπάρχουν οι βασικές λειτουργίες της εφαρμογής, οι οποίες είναι:

- Μετρητής Βημάτων
- Κυκλική μπάρα μέτρησης progress
- Μετρητής καμένων θερμίδων
- Μετρητής χιλιομέτρων
- Προσθήκη νερού
- Προσθήκη Βάρους

Στο Home Fragment ενεργοποιείται ο αισθητήρας “Accelerometer” ο οποίος είναι αυτός ο οποίος είναι υπεύθυνος για την καταμέτρηση των βημάτων. Αυτό επιτυγχάνεται με τη μέθοδο “onSensorChanged()” (Εικόνα 3.37).

- Αρχικά πρέπει να πάρουμε τις τιμές των δεδομένων των αξόνων X,Y και Z(Σχήμα 3.8) από τον αισθητήρα.

Κεφάλαιο 3

- Έπειτα υπολογίζουμε το συνολικό μέγεθος(Magnitude) των τριών διαστάσεων. Αυτό προκύπτει υπολογίζοντας το τετράγωνο των δεδομένων υψωμένων στο τετράγωνο, με την μαθηματική εξίσωση $\text{Sqrt}(x_acceleration^2 + y_acceleration^2 + z_acceleration^2)$ (Εικόνα 3.38)
- Μετά βρίσκουμε τη διαφορά από την προηγούμενη τιμή του Magnitude. Αν η διαφορά είναι μεγαλύτερη από την οριακή τιμή, τότε αυξάνουμε τα βήματα. Η οριακή τιμή για περπάτημα είναι >6 και για τρέξιμο >10 .

```
SensorEventListener stepDetector=new SensorEventListener() {
    @Override
    //Reading Events
    public void onSensorChanged(SensorEvent event) {
        if(event!=null){
            float x_acceleration = event.values[0];
            float y_acceleration = event.values[1];
            float z_acceleration = event.values[2];

            double Magnitude = Math.sqrt(x_acceleration*x_acceleration + y_acceleration*y_acceleration + z_acceleration*z_acceleration);
            double MagnitudeDelta = Magnitude-MagnitudePrevious;
            MagnitudePrevious = Magnitude;

            if (MagnitudeDelta > 6){
                stepCount++;
            }

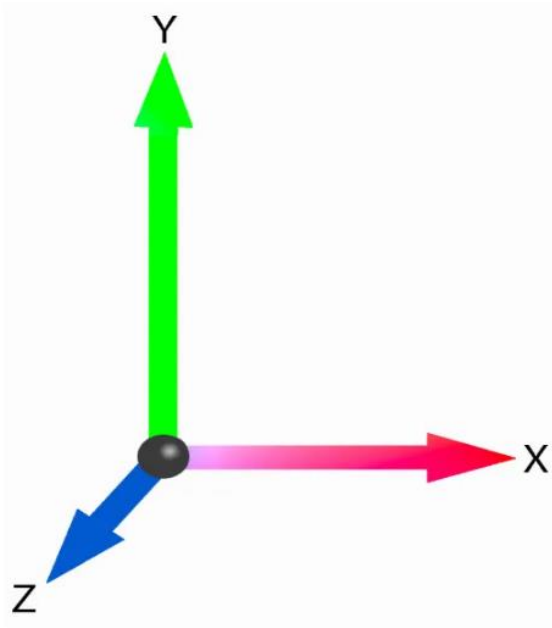
            steps.setText(stepCount.toString());

            //progress circle
            int progress=(int) ( ((double)stepCount / (double)stepsGoal)*100);
            progressBar.setProgress(progress);

            distance.setText( calculateDistance(stepCount));
            calories.setText( calculateCalories(stepCount));

            calor=calculateCalories(stepCount);
            caloriesInt=Integer.parseInt( calculateCalories(stepCount));
        }
    }
}
```

Εικόνα 3.37 Μέθοδος onSensorChanged()



Σχήμα 3.8 Άξονες συντεταγμένων XYZ [26]

```

SensorEventListener stepDetector=new SensorEventListener() {
    @Override
    //Reading Events
    public void onSensorChanged(SensorEvent event) {
        if(event!=null){
            float x_acceleration = event.values[0];
            float y_acceleration = event.values[1];
            float z_acceleration = event.values[2];

            double Magnitude = Math.sqrt(x_acceleration*x_acceleration + y_acceleration*y_acceleration + z_acceleration*z_acceleration);
            double MagnitudeDelta = Magnitude-MagnitudePrevious;
            MagnitudePrevious = Magnitude;

            if (MagnitudeDelta > 6){
                stepCount++;
            }
        }
    }
}

```

Εικόνα 3.38 Μαθηματική συνάρτηση για την εύρεση του Magnitude

Για την εισαγωγή των δεδομένων στη βάση πρέπει πρώτα να δημιουργηθεί σύνδεση (Εικόνα 3.39):

```

//database connection
db = FirebaseFirestore.getInstance();
firebaseAuth = FirebaseAuth.getInstance();

```

Εικόνα 3.39 Σύνδεση με τη βάση

Έπειτα πρέπει να ελεγχθεί αν υπάρχει το UserId στον πίνακα Records και αν υπάρχει με την μέθοδο put() προστίθεται το κάθε πεδίο στον πίνακα. (Εικόνα 3.40)

```

userID = firebaseAuth.getCurrentUser().getUid();
DocumentReference documentReferenceRecords = db.collection("Records").document(userID);
documentReferenceRecords.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
    //on success
    @Override
    public void onComplete(@NonNull Task<DocumentSnapshot> task) {
        DocumentSnapshot document = task.getResult();

        Map<String, Object> records = new HashMap<>();
        records.put("Steps", stepCount);
        records.put("Calories", caloriesInt);
        records.put("Distance", distanceKm);
        documentReferenceRecords.set(records);
        System.out.println(document);
    }
    //on failure
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.w( tag: "ERROR", msg: "Error getting documents.", e);
    }
});

```

Εικόνα 3.40 Εισαγωγή δεδομένων στη βάση

Τέλος, πρέπει να αναφερθεί ότι στις μεθόδους onResume(), onPause(), onStop() προστέθηκαν οι αντίστοιχες εντολές ώστε να μένουν ενημερωμένες οι τιμές των βημάτων, θερμίδων κλπ. (Εικόνα 3.41)

```

public void onStop() {
    super.onStop();

    SharedPreferences sharedPreferences = getActivity().getPreferences(Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear();
    editor.putInt("stepCount", stepCount);
    editor.putInt("calories", caloriesInt);
    editor.putFloat("distance", distanceKm);
    editor.apply();
}

public void onResume() {
    super.onResume();

    SharedPreferences sharedPreferences = getActivity().getPreferences(Context.MODE_PRIVATE);
    stepCount = sharedPreferences.getInt( key: "stepCount", defValue: 0);
    caloriesInt = sharedPreferences.getInt( key: "calories", defValue: 0);
    distanceKm=sharedPreferences.getFloat( key: "distance", defValue: 0);
}

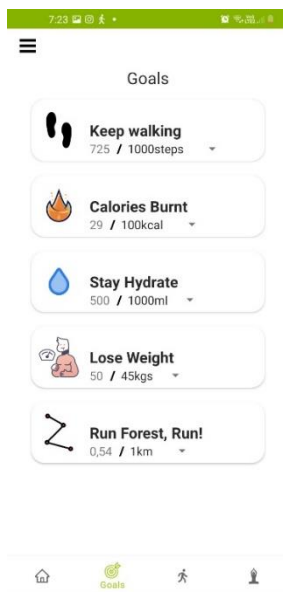
```

Εικόνα 3.41 Lifecycle μέθοδοι onResume(), onStop()

3.10.2 Goals Fragment

Το δεύτερο Fragment και η δεύτερη επιλογή του Bottom Menu είναι το Goals Fragment στο οποίο ο χρήστης μπορεί να δει τους ενδεικτικούς στόχους αλλά, και να θέσει τους δικούς του. Στο Goals Fragment υπάρχουν CardViews με ενδεικτικούς στόχους στην αρχή (Εικόνα 3.42). Τα CardView widgets είναι αντικείμενα της Material Design που αποδίδουν σχήμα μιας κάρτας με στρόγγυλες γωνίες και σκίαση, κάτι που δίνει την αίσθηση 3D αντικειμένου. Για να επιτραπεί η χρήση των CardViews πρέπει να προστεθεί η γραμμή implementation "androidx.cardview:cardview:1.0.0" στη κλάση dependencies build.gradle [23].

Ο χρήστης μπορεί από το αντίστοιχο Spinner να επιλέξει τον προσωπικό του στόχο όσον αφορά τα ημερήσια βήματα, την ημερήσια κατανάλωση νερού, τις ημερήσιες καμένες θερμίδες, τα κιλά και τέλος την απόσταση σε χιλιόμετρα που θέλει να διανύσει.



Εικόνα 3.42 Διεπαφή Goals

Για τη χρήση του Spinner πρέπει καταρχάς να δημιουργηθεί ένα αρχείο xml στα Resources το οποίο θα περιέχει ένα string-array (πίνακας χαρακτήρων) για κάθε spinner που θέλουμε να προσθέσουμε. Στη συνέχεια στο αντίστοιχο Layout xml αρχείο, προσθέτουμε spinner αντικείμενα (Εικόνα 3.43) για την κάθε κατηγορία στόχου. Στον κώδικα της Java εισάγουμε την βιβλιοθήκη Spinner, δημιουργούμε τα αντικείμενα κι έπειτα δημιουργούμε Array Adapters (Εικόνα 3.44) στους οποίους αντιστοιχούμε το κάθε array με τα spinner αντικείμενα του Layout xml. Για να δώσουμε το στυλ της “Drop down” λίστας-μενού, χρησιμοποιούμε την μέθοδο setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item). Τέλος, με τη μέθοδο setAdapter() θέτουμε τις τιμές του κάθε ArrayAdapter ως τις τιμές του κάθε Spinner της κλάσης. Στην Εικόνα 3.45 παρουσιάζεται η διεπαφή των Στόχων και ο Spinner για την θέση στόχου θερμίδων.

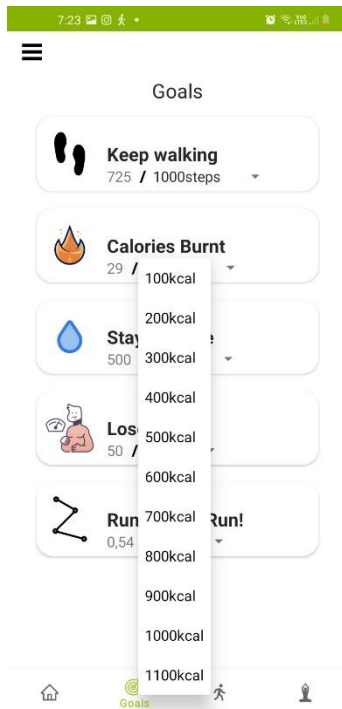
```
<Spinner
    android:id="@+id/spinnerCalories"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_below="@id/caloriesTxt"
    android:layout_toEndOf="@id/calsLine"
    android:layout_marginBottom="7dp"
    android:drawSelectorOnTop="true"/>
<Spinner
    android:id="@+id/spinnerWater"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_below="@id/waterGoaltxt"
    android:layout_toEndOf="@id/waterLine"
    android:layout_marginBottom="7dp"
    android:drawSelectorOnTop="true"/>
```

Εικόνα 3.43 Water & Calories Spinners XML

```
// Create an ArrayAdapter using the string array and a default spinner layout
ArrayAdapter<CharSequence> adapterCal = ArrayAdapter.createFromResource(getActivity(),
    R.array.caloriesGoals, android.R.layout.simple_spinner_item);
// Specify the layout to use when the list of choices appears
adapterCal.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinnerCalories.setAdapter(adapterCal);

ArrayAdapter<CharSequence> adapterMl = ArrayAdapter.createFromResource(getActivity(),
    R.array.watermlGoals, android.R.layout.simple_spinner_item);
adapterMl.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerWater.setAdapter(adapterMl);
```

Εικόνα 3.44 Spinner Java



Εικόνα 3.45 Γραφικό κομμάτι Spinner

3.10.3 Out Training Fragment

Το τρίτο Fragment και η τρίτη επιλογή του Bottom Menu είναι το Out Train Fragment στο οποίο ο χρήστης μπορεί να εκκινήσει κάποια αερόβια άσκηση, όπως το περπάτημα. Όταν επιλεγεί κάποια άσκηση, ανοίγει ένα νέο Activity που περιέχει τον χάρτη και ένα χρονόμετρο με τα κουμπιά Start και Reset. Σε αυτό το Fragment υπάρχουν οι επιλογές για τις αερόβιες ασκήσεις (περπάτημα κλπ.). Όταν επιλέξει ο χρήστης κάποια άσκηση, καλείται το Activity που περιέχει 2 άλλα Fragments, το ένα περιέχει ένα frame με τον χάρτη (Εικόνα 3.46) και το άλλο περιέχει ένα χρονόμετρο για να μετράει ο χρήστης την διάρκεια της άσκησης του, και τα κουμπιά “Start, Pause και Reset” (Εικόνα 3.46).



Εικόνα 3.46 Map Frame XML

Μόλις ανοίξει το Activity, εμφανίζεται η τοποθεσία του χρήστη στον χάρτη μόνο και εάν δοθεί αυτό το δικαίωμα από τον χρήστη. Αυτό ελέγχεται από τη μέθοδο `ActivityCompat.checkSelfPermission()`. (Εικόνα 3.47)

```
fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient( activity: this);
if (ActivityCompat.checkSelfPermission( context: OutTrainActivity.this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED)
    getCurrentLocation();
} else {
    ActivityCompat
        .requestPermissions( activity: OutTrainActivity.this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 44);
}
```

Εικόνα 3.47 Έλεγχος δικαιώματος τοποθεσίας

Αν τελικά δοθεί το δικαίωμα της παρούσας τοποθεσίας, προστίθεται «πινέζα» στον χάρτη και εστιάζεται εκείνη η περιοχή. Στην εικόνα 3.48 παρουσιάζεται η μέθοδος `onMapReady()` κατά την οποία γίνεται η τοποθέτηση πινέζας στην τοποθεσία του χρήστη, λαμβάνοντας τις συντεταγμένες Latitude και Longitude.

```
Task<Location> task = fusedLocationProviderClient.getLastLocation();
task.addOnSuccessListener(new OnSuccessListener<Location>() {
    @Override
    public void onSuccess(final Location location) {
        // when success
        if(location != null){
            // sync map
            mapFragment.getMapAsync(new OnMapReadyCallback() {
                @Override
                public void onMapReady(GoogleMap googleMap) {
                    //init lat lng

                    LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());
                    //create marker
                    MarkerOptions options= new MarkerOptions().position(latLng).title("");
                    //zoom at the current locantion on Map
                    googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));
                    //add marker on map
                    googleMap.addMarker(options);
                }
            });
        }
    }
});
```

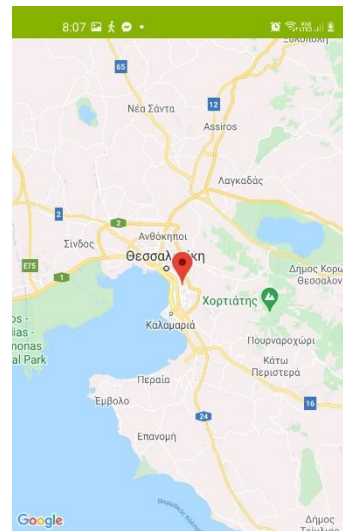
Εικόνα 3.48 Μέθοδος `onMapReady()`

Στην Εικόνα 3.49 φαίνεται ο χάρτης χωρίς να δοθεί το δικαίωμα της τοποθεσίας, γι' αυτό το λόγο δεν εμφανίζεται πινέζα. Στην Εικόνα 3.50, έχει επιτραπεί το δικαίωμα στην εφαρμογή να «δει» την τοποθεσία του χρήστη, κι έτσι εμφανίστηκε μια πινέζα στην τοποθεσία που βρίσκεται.



00:00:00

Εικόνα 3.49 Άρνηση παραχώρησης δικαιώματος τοποθεσίας



00:00:00

Εικόνα 3.50 Παραχώρηση δικαιώματος τοποθεσίας

Για το χρονοόμετρο πρέπει να προστεθούν αρχικά οι βιβλιοθήκες Timer και Timer task, να δημιουργηθούν τα αντίστοιχα αντικείμενα και έπειτα οι μέθοδοι έναρξης, παύσης και επαναφοράς. Έπειτα ελέγχεται με την μέθοδο startStopTapped() αν έχει πατηθεί το κουμπί έναρξης ή παύσης (Εικόνα 3.51). Για να έχει ο χρόνος τη σωστή μορφή σε ώρες, λεπτά και δευτερόλεπτα, πρέπει να δημιουργηθούν οι κλάσεις getTimeText() {} String formatTime() {} (Εικόνα 3.52). Για την επαναφορά του χρονομέτρου καλείται η μέθοδος resetTapped() (Εικόνα 3.53) στην οποία εμφανίζεται dialog παράθυρο για να επιβεβαιωθεί ότι όντως ο χρήστης θέλει να επαναφέρει το χρονοόμετρο. Αν επιλεγθεί το “Yes” ο χρόνος μηδενίζεται, αλλιώς το χρονοόμετρο συνεχίζει να τρέχει.

```
public void startStopTapped(View view)
{
    if(timerStarted == false)
    {
        timerStarted = true;
        setButtonUI( start: "STOP", R.color.magenta);

        startTimer();
    }
    else
    {
        timerStarted = false;
        setButtonUI( start: "START", R.color.purple_700);

        timerTask.cancel();
    }
}
```

Εικόνα 3.51 Μέθοδος αλλαγής κουμπιού Start-Stop

```

private void startTimer()
{
    timerTask = (TimerTask) () -> {
        runOnUiThread(new Runnable()
        {
            @Override
            public void run()
            {
                time++;
                timerText.setText(getTimerText());
            }
        });
};
timer.scheduleAtFixedRate(timerTask, delay: 0 , period: 1000);
}

private String getTimerText()
{
    int rounded = (int) Math.round(time);

    int seconds = ((rounded % 86400) % 3600) % 60;
    int minutes = ((rounded % 86400) % 3600) / 60;
    int hours = ((rounded % 86400) / 3600);

    return formatTime(seconds, minutes, hours);
}

private String formatTime(int seconds, int minutes, int hours)
{
    return String.format("%02d",hours) + " : " + String.format("%02d",minutes) + " : " + String.format("%02d",seconds);
}

```

Εικόνα 3.52 Μέθοδοι Start και Format Timer

```

public void resetTapped(View view)
{
    AlertDialog.Builder resetAlert = new AlertDialog.Builder( context: this);
    resetAlert.setTitle("Reset Timer");
    resetAlert.setMessage("Are you sure you want to reset the timer?");
    resetAlert.setPositiveButton( text: "Reset", new DialogInterface.OnClickListener()
    {
        @Override
        public void onClick(DialogInterface dialogInterface, int i)
        {
            if(timerTask != null)
            {
                timerTask.cancel();
                setButtonUI( start: "START", R.color.teal_200);
                time = 0.0;
                timerStarted = false;
                timerText.setText(formatTime( seconds: 0, minutes: 0, hours: 0));
            }
        }
    });

    resetAlert.setNegativeButton( text: "Cancel", new DialogInterface.OnClickListener()
    {
        @Override
        public void onClick(DialogInterface dialogInterface, int i)
        {
            //do nothing
        }
    });

    resetAlert.show();
}

```

Εικόνα 3.53 Reset timer

3.10.4 Stretch Fragment

Στο τελευταίο Fragment (Stretch Fragment Εικόνα 3.54) και επιλογή στο Bottom Menu, προβάλλονται εικόνες με σιλουέτες ανθρώπων οι οποίες δείχνουν κάποιες ασκήσεις προθέρμανσης και διατάσεων για μετά την άσκηση.



Εικόνα 3.54 Stretch Fragment

3.11 App resources

Τα resources είναι τα πρόσθετα αρχεία και το στατικό περιεχόμενο που χρησιμοποιεί ο κώδικας, όπως bitmap, προσδιορισμοί layout, user interface strings, οδηγίες κινούμενων σχεδίων και άλλα.

3.11.1 Menu

3.11.1.1 Bottom Menu

Όπως αναφέρθηκε και παραπάνω, για την καλύτερη πλοήγηση στην εφαρμογή, δημιούργησα ένα μενού το οποίο εμφανίζεται στο κάτω μέρος του “Home” Activity. Με αυτό το μενού γίνεται εύκολη η μετάβαση από Fragment σε Fragment (Εικόνες 3.55 και 3.56).

```

<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/nav_home"
    android:icon="@drawable/home"
    android:title="Home" />

  <item
    android:id="@+id/nav_goal"
    android:icon="@drawable/iconfinder_thefreeforty_target_1243705"
    android:title="Goals" />

  <item
    android:id="@+id/nav_run"
    android:icon="@drawable/walk"
    android:title="Run" />

  <item
    android:id="@+id/nav_stretch"
    android:icon="@drawable/yoga"
    android:title="Stretch" />
</menu>

```

Εικόνα 3.55 XML Bottom Menu items



Εικόνα 3.56 Γραφικό κομμάτι Bottom Menu

3.11.2 Drawer shift- Side Menu

Στο πάνω αριστερά μέρος του “Home” Activity, υπάρχει το λεγόμενο “Hamburger”, το οποίο πήρε αυτόν τον χαρακτηρισμό καθώς μοιάζει οπτικά με ένα burger. Κάνοντας κλικ σε αυτό το αντικείμενο-οντότητα εμφανίζεται το πλαϊνό μενού (Side Menu) και ανοίγει σαν «συρτάρι», εξού και η ονομασία «Drawer Shift» (Εικόνα 3.58). Απ’ αυτό μπορεί κανείς να περιηγηθεί σε εξτρά δυνατότητες όπως:

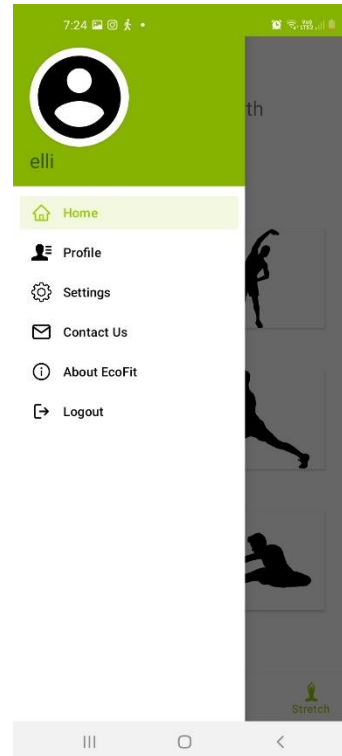
- Προβολή Προφίλ
- Ρυθμίσεις
- Φόρμα επικοινωνίας με τον δημιουργό της εφαρμογής
- Πληροφορίες
- Αποσύνδεση

```

<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single">
    <item
      android:id="@+id/nav_home"
      android:icon="@drawable/home"
      android:title="Home" />
    <item
      android:id="@+id/nav_profile"
      android:icon="@drawable/user"
      android:title="Profile" />
    <item
      android:id="@+id/nav_settings"
      android:icon="@drawable/settings"
      android:title="Settings" />
    <item
      android:id="@+id/nav_contact"
      android:icon="@drawable/mail"
      android:title="Contact Us" />
    <item
      android:id="@+id/nav_about"
      android:icon="@drawable/info"
      android:title="About EcoFit" />
    <item
      android:id="@+id/nav_logout"
      android:icon="@drawable/logout_line"
      android:title="Logout" />
  </group>
</menu>

```

Εικόνα 3.57 XML Drawer Shift Menu



Εικόνα 3.58 Γραφικό κομμάτι Drawer Shift Menu

Με τον ορισμό ως single στο μπλοκ “group”, δηλώνεται ότι μόνο μια επιλογή είναι δυνατή (Εικόνα 3.57). Για να «ανοίξει» το Drawer Shift χρησιμοποιείται η μέθοδος `navigationDrawer()` και για να κλείσει η μέθοδος `drawerLayout.closeDrawer(GravityCompat.START)`. (Εικόνα 3.59)

```

private void navigationDrawer() {
    navigationView.bringToFront();
    navigationView.setNavigationItemSelectedListener(this);
    navigationView.setCheckedItem(R.id.nav_home);

    menuIcon.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (drawerLayout.isDrawerVisible(GravityCompat.START))
                drawerLayout.closeDrawer(GravityCompat.START);
            else drawerLayout.openDrawer(GravityCompat.START);
        }
    });
}

```

Εικόνα 3.59 Εμφάνιση Drawer Shift

Η πλοήγηση στο Drawer Shift μενού γίνεται μέσω της μεθόδου `onNavigationItemSelectedListener(@NonNull MenuItem item)`. (Εικόνα 3.60)

```
public boolean onNavigationItemSelectedListener(@NonNull MenuItem item) {
    int id = item.getItemId();
    Fragment selectedFragment=null;
    switch (id) {
        case R.id.nav_home:
            selectedFragment=new HomeFragment();
            break;
        case R.id.nav_profile:
            selectedFragment=new ProfileFragment();
            break;
        case R.id.nav_settings:
            selectedFragment=new SettingsFragment();
            break;
        case R.id.nav_contact:
            //selectedFragment=new ContactFragment();
            Intent intentContact = new Intent( packageContext: HomeActivity.this, ContactUsActivity.class);
            startActivity(intentContact);
            finish();
            break;
        case R.id.nav_logout:
            firebaseAuth.signOut();
            Intent intent = new Intent( packageContext: HomeActivity.this, LoginActivity.class);
            startActivity(intent);
            finish();
            break;
    }
    drawerLayout.closeDrawer(GravityCompat.START);
    if (selectedFragment!=null) {
        getSupportFragmentManager().beginTransaction().replace(R.id.container, selectedFragment).commit();
    }
    return true;
}
```

Εικόνα 3.60 Μέθοδος `onNavigationItemSelectedListener()`

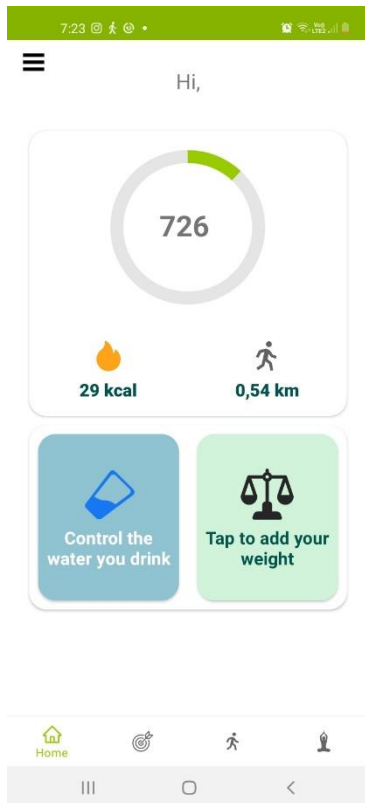
3.11.3 Progress Bar

Για να δημιουργήσω τον κύκλο ο οποίος γεμίζει όσο αυξάνονται τα ημερήσια βήματα, χρησιμοποίησα ένα Progress Bar. Είναι ένα στοιχείο διεπαφής χρήστη που δείχνει την πρόοδο μιας λειτουργίας. Η γραμμή προόδου- Progress Bar υποστηρίζει δύο τρόπους για την αναπαράσταση της προόδου: καθορισμένη και απροσδιόριστη.

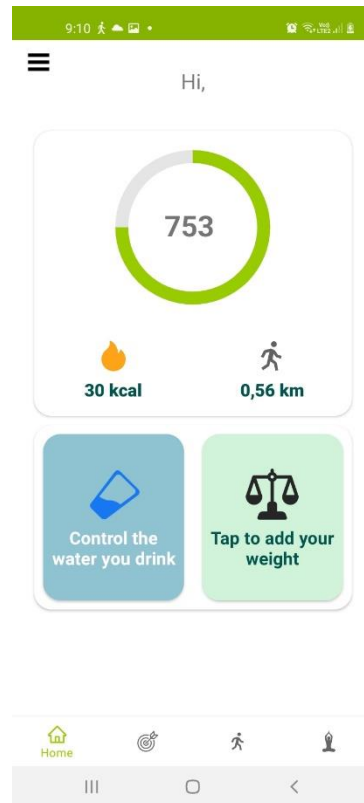
Η απροσδιόριστη λειτουργία χρησιμοποιείται όταν δεν γνωρίζουμε πόσο καιρό θα χρειαστεί μια λειτουργία για να ολοκληρωθεί. Η απροσδιόριστη λειτουργία είναι η προεπιλεγμένη γραμμή προόδου και εμφανίζει κυκλική κίνηση χωρίς να υποδεικνύεται συγκεκριμένη ποσότητα προόδου. Η καθορισμένη λειτουργία χρησιμοποιείται όταν θέλουμε να δείξουμε ότι έχει ολοκληρωθεί μια συγκεκριμένη ποσότητα προόδου. Για παράδειγμα, το ποσοστό αρχείου που έχει απομείνει από ένα αρχείο που ανακτάται. Στην εφαρμογή μου χρησιμοποιείται η καθορισμένη (Εικόνα 3.61) καθώς θέλω να αναπαραστήσω το ποσοστό βημάτων από τον ημερήσιο στόχο βημάτων έχουν γίνει. Στην Εικόνα 3.62 παρουσιάζεται το Progress Bar με Step Goal=6000 ενώ στην Εικόνα 3.63 έχει αλλάξει ο στόχος στα 1000 βήματα.

```
//progress circle
int progress=(int) ( ((double)stepCount / (double)stepsGoal) *100);
progressBar.setProgress(progress);
```

Εικόνα 3.61 Μέθοδος `setProgress()`



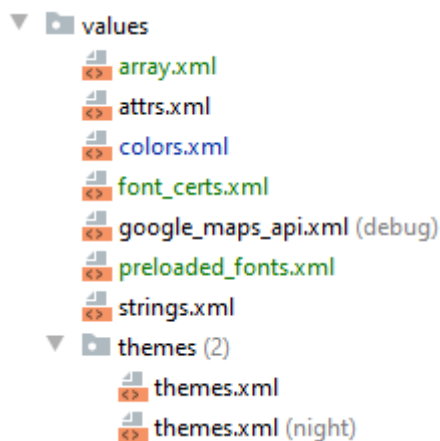
Εικόνα 3.62 Stepgoal=6000



Εικόνα 3.63 Stepgoal=1000

3.11.3.1 Values resources

Στην Εικόνα 3.64 βλέπουμε τα resources της εφαρμογής.



Εικόνα 3.64 Values resources

Array: Στο αρχείο **array.xml** περιέχονται οι πίνακες χαρακτήρων (string-array) που χρησιμοποιούνται στους Spinner για να πάρουν τιμές.

Στο αρχείο colors έχω δημιουργήσει μεταβλητές χρωμάτων στις οποίες έχω δώσει τιμές του δεκαεξαδικού, έτσι ώστε να χρησιμοποιώ την ονομασία των χρωμάτων χωρίς να ψάχνω δεκαεξαδικούς χαρακτήρες κάθε φορά που θέλω να χρησιμοποιήσω ένα χρώμα. (Εικόνα 3.65)

```
<color name="ecofitDarker">#86b300</color>  
<color name="ecofitLighter">#c6ff1a</color>  
<color name="ecofitComplementary">#3300cc</color>  
<color name="gray">#e5e5e5</color>  
<color name="grayDarker">#262626</color>  
<color name="babyBlue">#9573F1F1</color>  
<color name="babyGreenn">#9C9FC770</color>  
<color name="lemon">#A9F8D876</color>  
<color name="redPastel">#A8FF3526</color>  
<color name="orangee">#F6B093</color>  
<color name="banana">#fff8de</color>  
<color name="greenBlue">#1AB8A3</color>  
<color name="greenBlueDark">#01554E</color>
```

Εικόνα 3.65 Colors

Στο αρχείο **google_maps_api.xml** πρέπει να προστεθεί το προσωπικό API key του προγραμματιστή της εφαρμογής έτσι ώστε να μπορεί να χρησιμοποιήσει τους χάρτες της Google.

Στο αρχείο **strings** περιέχονται οι συμβολοσειρές με των API keys (facebook, google maps κλπ.).

Κεφάλαιο 4ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Μετά από την έρευνα και την προσωπική εμπειρία, θεωρώ ότι μια εφαρμογή για την σωματική άσκηση είναι αρκετά χρήσιμη, σε καιρό καραντίνας και όχι μόνο. Η έρευνα, τα αποτελέσματα της οποίας παρουσιάστηκαν στην αρχή του 1^{ου} κεφαλαίου, βοήθησε στην ανάπτυξη μιας χρήσιμης εφαρμογής σύμφωνα με τις ανάγκες των χρηστών και η μελέτη περί UX/UI βοήθησε στο να γίνει μια εφαρμογή φιλική και εύχρηστη από άτομα που δεν είναι τόσο εξοικειωμένα με την τεχνολογία. Η χρήση της Firebase έθεσαν την σχεδίαση της βάσης ευκολότερη από SQL βάσεις, κι επίσης το ότι είναι διαδικτυακή ήταν αρκετά χρήσιμο καθώς κανείς μπορεί να έχει πρόσβαση σε αυτή απομακρυσμένα και εκτός του προσωπικού του υπολογιστή. Η ανάπτυξη μιας Android εφαρμογής στο Android Studio από κάποιον προγραμματιστή με γνώσεις της γλώσσας java, είναι αρκετά απλή, χρησιμοποιώντας βέβαια και τις κατάλληλες βιβλιοθήκες που παρέχονται δωρεάν. Οι αισθητήρες που πλέον έχουν όλα τα κινητά Android εξυπηρετούν στο χτίσιμο μιας πιο διαδραστικής εφαρμογής χωρίς να υπάρχει ανάγκη έξτρα εξοπλισμού, παραδείγματος χάρη έξυπνο ρολόι.

Στόχος είναι να βελτιωθούν και να προστεθούν κάποια πράγματα στο μέλλον, όπως η προσθήκη 3D animation με φιγούρες οι οποίες θα δείχνουν τις ασκήσεις, κι έπειτα με την βοήθεια αισθητήρων θα ελέγχεται αν ο χρήστης κάνει σωστά την άσκηση. Επιπλέον θα ήθελα να κάνω δυνατό το χτίσιμο ενός κοινωνικού δικτύου με τους χρήστες της εφαρμογής, έτσι ώστε να δοθεί η δυνατότητα του διαμοιρασμού των στοιχείων των χρηστών και της σύγκρισης ρεκόρ με άλλους χρήστες της εφαρμογής.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <http://www.chemistry.wustl.edu/> Rachel Casiday, Greg Noelken, and Regina Frey “Treating the Public Water Supply: What Is In Your Water, and How Is It Made Safe to Drink?” July 2005 (Access 5/21)
- [2] <https://www.cdc.gov/chronicdisease/> (Accessed 6/21)
- [3] <https://kotlinlang.org/> (Accessed 5/21)
- [4] “Firebase docs”, *Firebase*. [Online]. Available: <https://firebase.google.com/> . (Accessed 5/21)
- [5] Chao Wang, Wei Duan, Jianzhang Ma and Chenhui Wang, "The research of Android System architecture and application programming," Proceedings of 2011 International Conference on Computer Science and Network Technology, 2011, pp. 785-790, doi: 10.1109/ICCSNT.2011.6182081.
- [6] Online Map Application Development Using Google Maps API, SQL Database, and ASP.NET
- [7] Introducing Google Chart Tools and Google Maps API in Data Visualization Courses
- [8] <https://rapidapi.com/blog/top-map-apis/> (Accessed 5/21)
- [9] <https://developers.facebook.com/>(Accessed 6/21)
- [10] K. Kuusinen, and T. Mikkonen, “On Designing UX for Mobile Enterprise Apps,” 40th Euromicro Conference on Software Engineering and Advanced Applications, 2014.
- [11] <https://www.behance.net/> (Accessed 5/21)
- [15] M. Christie. The psychology of logo shapes: A designer's guide [Online] from <https://www.creativebloq.com/logo-design/psychology-logo-shapes-8133918> (2017) [Accessed on 6/21].
- [16] Mullins, C. (2015). Responsive, mobile app, mobile first. Proceedings of the 33rd Annual International Conference on the Design of Communication - SIGDOC '15. doi:10.1145/2775441.2775478
- [17] <https://www.figma.com/> (Accessed 5/21)
- [18] Tejas Vithani and Anand Kumar(2014) ,”Modeling the Mobile Application Development Lifecycle”,Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol I,IMECS 2014, March 12 - 14, Hong Kong
- [19] Kaur, Anureet & Kaur, Kulwant. (2015). Suitability of Existing Software Development Life Cycle (SDLC) in Context of Mobile Application Development Life Cycle (MADLC). International Journal of Computer Applications. 116. 1-6. 10.5120/20441-2785.
- [20] <https://material.io/design/introduction> (Accessed 5/21)
- [21] <https://github.com/airbnb/lottie-android> lottie animation (Access 5/21)
- [22] <https://github.com/koral--/android-gif-drawable> (Access 5/21)
- [23] *Android Developers*. [Online]. Available: <https://developers.google.com>. (Accessed 5/21)

- [24] F. Alswaina and K. Elleithy, "Android Malware Family Classification and Analysis: Current Status and Future Directions," *Electronics*, vol. 9, no. 6, p. 942, Jun. 2020.
- [25] Evis Canziba, "Hands-On UX Design for Developers: Design, prototype, and implement compelling user experiences from scratch." Oct. 2018.
- [26] <https://programmerworld.co/android/how-to-create-walking-step-counter-app-using-accelerometer-sensor-and-shared-preference-in-android/> (Access 5/21)
- [27] "Understand the Activity Lifecycle", *Android Developers*. [Online]. Available: <https://developer.android.com/guide/components/activities/activity-lifecycle> . (Accessed 5/21)
- [28] J. Chen, G. Han, S. Guo and W. Diao, "FragDroid: Automated User Interface Interaction with Activity and Fragment Analysis in Android Applications," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2018, pp. 398-409, doi: 10.1109/DSN.2018.00049.
- [29] Junior Cupe Casquina, Jane D. A. Sandim Eleuterio and Cecilia M. F. Rubira "Adaptive Deployment Infrastructure for Android Applications" Conference: 2016 12th European Dependable Computing Conference (EDCC), Year: 2016, Page 21 doi: 10.1109/EDCC.2016.25
- [30] https://www.adobe.com/gr_en/products/aftereffects.html (Accessed 6/21)
- [31] <https://blog.adobe.com/en/publish/2017/11/29/prototyping-difference-low-fidelity-high-fidelity-prototypes-use.html#gs.3trwuy> (Accessed 6/21)
- [33] Soo Yun Kim, Youn Lee "Using High Fidelity Interactive Prototypes for Effective Communication to Create an Enterprise Mobile Application" CIO Design, IBM, New York, NY, USA
- [34] Friskila Enggar Pamudyaningrum¹, Hestiasari Rante¹, Muhammad Agus Zainuddin and Michael Lund "UI/UX Design for Metora: A Gamification of Learning Journalism Interviewing Method", September 2020
- [35] <https://github.com/hdodenhof/CircleImageView> (Accessed 6/21)
- [36] <https://www.apple.com/ios/> (Accessed 6/21)
- [37] <https://flutter.dev/> (Accessed 6/21)
- [38] <https://exchange.adobe.com/creativecloud.details.12557.bodymovin.html> (Accessed 6/21)
- [39] <https://pixabay.com/users/firmbee-663163/> (Accessed 6/21)