



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

**Σχεδίαση και Ανάπτυξη Συστήματος Αυτοματοποιημένης Επίβλεψης και
Καθοδήγησης Δρομέων**



**Φοιτητής: Ματέους Κσιστόφ Ματζιάρ
ΑΜ: 50106448**

Επιβλέπων:

**Ξενοφών Διονύσιος Κανδρής
Καθηγητής**

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΜΑΡΤΙΟΣ 2023



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Design and Development of an Automated System for the Supervision and Guidance of Runners



Student: Mateusz Krzysztof Madziar
Registration Number: 50106448

Supervisor

Xenofon Dionisis Kandris
Professor

ATHENS-EGALEO, MARCH 2023

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Ξενοφών Διονύσιος Κανδρής Καθηγητής	Γρηγόριος Κουλούρας, Αναπληρωτής Καθηγητής	Ηλίας Ζώης, Αναπληρωτής Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Ματέους Κσιστόφ Ματζιάρ, Μάρτιος, 2023

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ματζιάρ Ματέους Κσιστόφ του Κσιστόφ, με αριθμό μητρώου 50106448 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.»

Ο Δηλών
Ματέους Κσιστόφ Ματζιάρ



Ευχαριστίες

Με την παρούσα διπλωματική εργασία ολοκληρώνονται οι σπουδές μου στο προπτυχιακό πρόγραμμα σπουδών του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών του Πανεπιστημίου Δυτικής Αττικής.

Το μεγαλύτερο ευχαριστώ, το οφείλω στην οικογένεια μου για τη διαρκή υποστήριξη και τη βοήθεια όλα τα χρόνια που διένυσα ως προπτυχιακός φοιτητής, αλλά και καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή κ. Διονύση Κανδρή για την καθοδήγηση του σε όλο το διάστημα έρευνας, μελέτης και υλοποίησης της διπλωματικής εργασίας, παρέχοντας μου πολύτιμες συμβουλές και γνώσεις.

Επιπλέον, θα ήθελα να ευχαριστήσω όλους τους κοντινούς μου ανθρώπους, για τη συμπαράσταση τους όλο αυτό το διάστημα.

Περίληψη

Η διπλωματική εργασία περιγράφει τη δομή και τη λειτουργία εφαρμογής παρακολούθησης και καθοδήγησης αθλητών η οποία έχει ως στόχο τη μεγιστοποίηση της απόδοσης του αθλούμενου και συγχρόνως, τη μείωση της πιθανότητας υπερκόπωσης. Δίνεται έμφαση στο ψυχολογικό προφίλ του αθλούμενου, αλλά και στην αίσθηση του πριν και κατά τη διάρκεια της προπόνησης.

Το τεστ προσωπικότητας είναι βασισμένο στο "Big Five Inventory model" και παρέχει όλες τις απαραίτητες πληροφορίες για την προσωπικότητα του αθλητή. Επιπλέον, δίνει στον προπονητή μια ολοκληρωμένη εικόνα για την κατάσταση του αθλητή κατά τη διάρκεια της προπόνησης και του επιτρέπει την τροποποίηση του προπονητικού προγράμματος σε πραγματικό χρόνο. Δίνει στον αθλητή τη δυνατότητα επικοινωνίας με όλη την ομάδα που συντελεί στην επίτευξη του καλύτερου δυνατού αποτελέσματος.

Επίσης, παρέχει συμβουλές διατροφής προσαρμοσμένες στις ενεργειακές απαιτήσεις της καθημερινότητας του αθλούμενου σε συνάρτηση με την προπονητική επιβάρυνση.

Η εφαρμογή υλοποιήθηκε με τη χρήση του προγράμματος Android Studio και η γλώσσα προγραμματισμού που επιλέχθηκε είναι η Kotlin. Είναι συμβατή με το 98% όλων των κινητών τηλεφώνων που χρησιμοποιούν λογισμικό Android.

Λέξεις – κλειδιά

Συστήματα Επίβλεψης Απόδοσης Αθλητών, Εφαρμογή Android, Android Studio, Γλώσσα Προγραμματισμού Kotlin, Εφαρμογές κινητών τηλεφώνων, Big Five Inventory.

Abstract

This thesis describes the structure and the functionality of a mobile application that is responsible for monitoring and evaluating an athlete's performance. It aims to maximize the athlete's performance while simultaneously decreasing the probability of overload. Emphasis is placed on the user's personality profile and their answers before and during the workout.

Contains a personality test based on the "Big Five Inventory model" which provides the necessary information about the athlete's personality.

Additionally, provides a complete report about the athlete's performance and their feelings during the training. It allows the coach to make edits in real-time on the current training plan.

Moreover, the application allows the athlete to communicate with the whole team, which contributes to achieving the best possible result.

It also provides nutrition advice adapted to the energy requirements of the athlete's daily routine in relation to the training load.

The application was created by using Android Studio and the chosen programming language was Kotlin. It is compatible with 98% of all mobile phones using Android software.

Keywords

Athlete Performance Evaluation Systems, Android Application, Android Studio, Programming Language Kotlin, Mobile Development, Big Five Inventory.

Περιεχόμενα

Κατάλογος Πινάκων.....	12
Κατάλογος Εικόνων	12
Αλφαβητικό Ευρετήριο.....	16
ΕΙΣΑΓΩΓΗ.....	17
Αντικείμενο της διπλωματικής εργασίας	17
Σκοπός και στόχοι	17
Μεθοδολογία.....	18
Καινοτομία.....	18
Δομή.....	19
1 ΚΕΦΑΛΑΙΟ 1^ο : Θεωρητικό Υπόβαθρο.....	20
1.1 Εισαγωγή κεφαλαίου	20
1.2 Παράγοντες που συμβάλουν στην απόδοση αθλητή	20
1.2.1 Βασικές αρχές της προπονητικής	20
1.2.2 Διατροφή και ενεργειακές απαιτήσεις	21
1.2.3 Σημασία της ξεκούρασης - αποκατάστασης.....	23
1.2.4 Ψυχολογικός παράγοντας	24
1.2.5 Εξωτερικοί παράγοντες.....	24
1.3 Βασικές έννοιες στη προπονητική	25
1.3.1 Ενεργειακά συστήματα του ανθρώπου	25
1.3.2 Αερόβια αντοχή	26
1.3.3 Αερόβιο κατώφλι	26
1.3.4 Αναερόβια αντοχή	27
1.3.5 Αναερόβιο κατώφλι	27
1.3.6 Μέγιστη πρόσληψη οξυγόνου VO ₂ _max	27
1.3.7 Ζώνες καρδιακών παλμών	28
1.4 Προσδιορισμός Ψυχολογικού Προφίλ του Αθλούμενου	29
1.4.1 Σημασία του ψυχολογικού προφίλ και η σχέση του με την απόδοση αθλητή	29
1.4.2 Big Five Inventory - BFI	29
1.4.3 Άλλα Ψυχολογικά τεστ	32
1.5 Συστήματα παρακολούθησης απόδοσης αθλητή	33
2 ΚΕΦΑΛΑΙΟ 2^ο : Τεχνικό υπόβαθρο	34
2.1 Εισαγωγή κεφαλαίου	34
2.2 Γλώσσα προγραμματισμού Kotlin	34
2.2.1 Εισαγωγή	34
2.2.2 Δημοφιλείς εφαρμογές που χρησιμοποιούν την Kotlin	35
2.3 Android Studio.....	36
2.3.1 Εισαγωγή	36
2.3.2 Δομή του Android Studio	36
2.3.3 Εφαρμογές Android για την παρακολούθηση αθλητικής δραστηριότητας	39
2.4 Βάση δεδομένων Firebase	39
2.4.1 Εισαγωγή – RealtimeDatabase	39
2.4.2 Σύνδεση με το Android Studio	40
2.4.3 Επαλήθευση χρήστη - User Authentication.....	42
2.4.4 Ανάκτηση και μεταφορά δεδομένων.....	43
2.4.5 Ασφάλεια της βάσης δεδομένων – Security Rules	43

3	ΚΕΦΑΛΑΙΟ 3^ο : Ανάπτυξη Εφαρμογής	45
3.1	Εισαγωγή κεφαλαίου	45
3.2	Δομή της εφαρμογής	45
3.3	Βασικές λειτουργίες της εφαρμογής	48
3.4	Ανάπτυξη εφαρμογής με τη χρήση Android Studio	49
3.4.1	Διαμόρφωση του Build Grandle File	49
3.4.2	Δραστηριότητα εισόδου (Welcome Activity)	51
3.4.2.1	Περιγραφή λειτουργείας	51
3.4.2.2	Animaton	52
3.4.3	Σύνδεση χρήστη	53
3.4.3.1	Περίληψη	53
3.4.3.2	Είσοδος με τη χρήση κωδικού πρόσβασης	54
3.4.3.3	Είσοδος με τη χρήση λογαριασμού Google	56
3.4.4	Δημιουργία νέου λογαριασμού	59
3.4.4.1	Εγγραφή νέου χρήστη	59
3.4.4.2	Συμπλήρωση στοιχείων νέου χρήστη	61
3.4.5	Επιλογή ανάκτησης κωδικού πρόσβασης	62
3.4.6	Κεντρική δραστηριότητα	64
3.4.7	Γενικές πληροφορίες για τον τρόπο λειτουργίας της εφαρμογής	65
3.4.8	Προφίλ αθλούμενου	66
3.4.9	Προσδιορισμός ψυχολογικού προφίλ	68
3.4.10	Start Training Acitvity	71
3.4.10.1	Περίληψη	71
3.4.10.2	Εβδομαδιαίο πρόγραμμα προπόνησης	72
3.4.10.3	Ζώνες καρδιακών παλμών και υπολογισμός ρυθμού	73
3.4.10.4	Επισκόπηση δραστηριότητας δρομέα	74
3.4.10.5	Εκκίνηση ημερίσιας δραστηριότητας	76
3.4.10.5.1	Περίληψη	76
3.4.10.5.2	Ερωτήσεις πριν την προπόνηση	76
3.4.10.5.3	Συμβουλές για άμεσες αλλαγές	77
3.4.10.5.4	Pause Activity	78
3.4.10.5.5	Αποθήκευση προπόνησης	79
3.4.10.5.6	Ερωτήσεις μετά την προπόνηση	81
3.4.10.5.7	Συμβουλές για την επόμενη προπόνηση	81
3.4.11	Επικοινωνία με την ομάδα υποστήριξης	82
3.4.11.1	Περίληψη	82
3.4.11.2	Δυνατότητα αποστολής με χρήση ηλεκτρονικού ταχυδρομείου	82
3.4.11.3	Δυνατότητα πραγματοποίησης κλήσης	84
3.4.12	Υπολογισμός θερμιδικών απαιτήσεων	85
3.4.12.1	Περιγραφή λειτουργίας	85
3.4.12.2	Πρόσθετες επιλογές I – Επικοινωνία με διατροφολόγο	86
3.4.12.3	Πρόσθετες επιλογές II – Εβδομαδιαίο πρόγραμμα διατροφής	87
3.4.12.4	Πρόσθετες επιλογές III –Βασικές πληροφορίες διατροφής	88
3.4.13	Είσοδος προπονητή	89
3.4.13.1	Επισκόπηση προπονητικού προγράμματος	89
3.4.13.2	Επιλογή μοφοποίησης προπονητικού προγράμματος	90
3.4.13.3	Επίβλεψη δραστηριότητας επιλεγμένου αθλητή	90
3.4.13.4	Πληροφορίες για τα αποτελεσμάτα ψυχολογικού τεστ	91
3.4.14	Επιπλέον λειτουργείες -Tool bar	93
3.4.15	Ασφάλεια δεδομένων - Security Rules	94
4	ΚΕΦΑΛΑΙΟ 4^ο : Επίλογος	95
4.1	Εισαγωγή κεφαλαίου	95

4.2	Σύνοψη Εργασίας.....	95
4.3	Προβλήματα – Επίλυση	96
4.4	Παρατηρήσεις – Συμπεράσματα	96
4.5	Προτάσεις Μελλοντικής Εξέλιξης	97
Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές		98

Κατάλογος Πινάκων

Πίνακας 1.1 Ανθρώπινα ενεργειακά συστήματα [38]	26
Πίνακας 1.2 Κλίμακα - Big Five Inventory [20]	30

Κατάλογος Εικόνων

Εικόνα 1.1 Οι πέντε μεγάλοι παράγοντες (Big Five Factors) [20]	30
Εικόνα 1.2 Ερωτήσεις Big Five Inventory [20]	31
Εικόνα 1.3 Υπολογισμός βαθμολογίας (Big Five Factors) [20]	32
Εικόνα 1.4 Garmin Forerunner 945 I [44]	33
Εικόνα 1.5 Garmin Forerunner 945 II [43]	33
Εικόνα 1.6 Εικόνα Catapult 1 [42]	33
Εικόνα 1.7 Εικόνα Catapult 2 [43]	33
Εικόνα 2.1 Activity Lifecycle [15]	37
Εικόνα 2.2 Ελάχιστες προδιαγραφές εφαρμογής [35]	41
Εικόνα 2.3 Σύνδεση Firebase – Απαραίτητος κώδικας I [35]	41
Εικόνα 2.4 Σύνδεση Firebase – Απαραίτητος κώδικας II [35]	41
Εικόνα 2.5 Σύνδεση Firebase – Απαραίτητος κώδικας III [35]	42
Εικόνα 2.6 Ανάκτηση δεδομένων απο τη Realtime Database [33]	43
Εικόνα 2.7 Εγγραφή δεδομένων στη Realtime Database [33]	43
Εικόνα 3.1 Δομή της εφαρμογής.....	45
Εικόνα 3.2 Δομή βάσης δεδομένων	45
Εικόνα 3.3 Ερωτήσεις πριν την προπόνηση	46
Εικόνα 3.4 Ερωτήσεις μετά την προπόνηση	46
Εικόνα 3.5 Ομάδα υποστήριξης	46
Εικόνα 3.6 Πρόγραμμα προπόνησης I	47
Εικόνα 3.7 Πρόγραμμα προπόνησης II	47
Εικόνα 3.8 Αποθήκευση στοιχείων χρήστη	47
Εικόνα 3.9 Βασικά στοιχεία αθλητή	48
Εικόνα 3.10 View Binding	49

<i>Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων</i>	
Εικόνα 3.11 Plugins.....	50
Εικόνα 3.12 Dependencies	50
Εικόνα 3.13 Πρώτη Δραστηριότητα	51
Εικόνα 3.14 Μεταφορά στην κεντρική δραστηριότητα εισόδου	51
Εικόνα 3.15 Δήλωση της πρώτης δραστηριότητας στο Manifest File.....	51
Εικόνα 3.16 Splash Animation.....	52
Εικόνα 3.17 Λειτουργία δραστηριότητας εισόδου.....	52
Εικόνα 3.18 Login Activity	53
Εικόνα 3.19 Βιβλιοθήκες Login Activity.....	53
Εικόνα 3.20 Σύνδεση αρχείου xml με το αρχείο Kotlin – Login Activity	54
Εικόνα 3.21 Μεταφορά στην κεντρική δραστηριότητα	54
Εικόνα 3.22 Έλεγχος κενής εισόδου	55
Εικόνα 3.23 Επαλήθευση στοιχείων χρήστη.....	55
Εικόνα 3.24 Διαμόρφωση Google Sign-in Button - Layout	56
Εικόνα 3.25 Επιλογή λογαριασμού Google	56
Εικόνα 3.26 Συνάρτηση registerActivityForGoogleSignIn().....	57
Εικόνα 3.27 Συνάρτηση firebaseSignInWithGoogle().....	57
Εικόνα 3.28 Συνάρτηση firebaseGoogleAccount().....	58
Εικόνα 3.29 Λειτουργία Google Sign-in Button.....	58
Εικόνα 3.30 Συνάρτηση signInGoogle()	58
Εικόνα 3.31 Συνάρτηση signIn()	58
Εικόνα 3.32 SignUp Activity	59
Εικόνα 3.33 Διαμόρφωση του Sign Up Button	59
Εικόνα 3.34 Δημιουργία νέου λογαριασμού στη πλατφόρμα Firebase	60
Εικόνα 3.35 Εισαγωγή βασικών πληροφοριών του αθλητή	61
Εικόνα 3.36 Μεταφορά δεδομένων στη βάση δεδομένων, δημιουργία database Reference.....	61
Εικόνα 3.37 Συνάρτηση giveFirstValuesWithPersonalInfo().....	61
Εικόνα 3.38 Δραστηριότητα ανάκτησης κωδικού πρόσβασης	62
Εικόνα 3.39 Διαμόρφωση του κουμπιού ανάκτησης κωδικού	62

<i>Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων</i>	
Εικόνα 3.40 Αλλαγή κωδικού πρόσβασης I	63
Εικόνα 3.41 Αλλαγή κωδικού πρόσβασης II	63
Εικόνα 3.42 Αλλαγή κωδικού πρόσβασης III	63
Εικόνα 3.43 Ρυθμίσεις μηνύματος για ανάκτηση κωδικού πρόσβασης.....	63
Εικόνα 3.44 Κεντρική δραστηριότητα	64
Εικόνα 3.45 Στοιχεία χρήστη	64
Εικόνα 3.46 Έλεγχος στοιχείων χρήστη.....	64
Εικόνα 3.47 Δραστηριότητα πληροφοριών.....	65
Εικόνα 3.48 Γενικές πληροφορίες – αρχείο «String.xml»	65
Εικόνα 3.49 Προφίλ Αθλούμενου	66
Εικόνα 3.50 Εμφάνιση δεδομένων αθλητή	67
Εικόνα 3.51 Αποθήκευση αλλαγών στο προφίλ αθλητή.....	67
Εικόνα 3.52 Ψυχολογικό Προφίλ.....	68
Εικόνα 3.53 Συνάρτηση PersonalityTestLogic()	68
Εικόνα 3.54 Υπολογισμός παραμέτρων I.....	69
Εικόνα 3.55 Υπολογισμός παραμέτρων II	69
Εικόνα 3.56 Υπολογισμός παραμέτρων III	69
Εικόνα 3.57 Υπολογισμός παραμέτρων IV	69
Εικόνα 3.58 Υπολογισμός παραμέτρων V	70
Εικόνα 3.59 Αποθήκευση αποτελέσματος ψυχολογικού προφίλ.....	70
Εικόνα 3.60 Start Training Activity	71
Εικόνα 3.61 Εβδομάδες Προπόνησης I.....	72
Εικόνα 3.62 Εβδομάδες Προπόνησης II.....	72
Εικόνα 3.63 Εβδομάδες Προπόνησης III	72
Εικόνα 3.64 Ζώνες καρδιακών παλμών	73
Εικόνα 3.65 Ρυθμοί προπόνησης	73
Εικόνα 3.66 Επιλογή αναζήτησης δεδομένων I	74
Εικόνα 3.67 Επιλογή αναζήτησης δεδομένων II	75
Εικόνα 3.68 Επιλογή αναζήτησης δεδομένων III	75

<i>Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων</i>	
Εικόνα 3.69 Διάγραμμα Start Training Activity	76
Εικόνα 3.70 Ερωτήσεις πριν την προπόνηση.....	77
Εικόνα 3.71 Συμβουλές πριν την προπόνηση	78
Εικόνα 3.72 PauseActivity	78
Εικόνα 3.73 Καταγραφή δραστηριότητας και αποστολή δεδομένων	79
Εικόνα 3.74 Αποθήκευση προπονήσεων στη βάση δεδομένων.....	80
Εικόνα 3.75 Ερωτήσεις μετά την προπόνηση	81
Εικόνα 3.76 Συμβουλές για την επόμενη ημέρα	81
Εικόνα 3.77 Ομάδα υποστήριξης I.....	82
Εικόνα 3.78 Ομάδα υποστήριξης II	82
Εικόνα 3.79 Αποστολή μηνύματος με τη χρήση ηλεκτρονικού ταχυδρομείου	83
Εικόνα 3.80 Συνάρτηση SendEmail()	83
Εικόνα 3.81 Απαραίτητη ρύθμιση στο Manifest File	84
Εικόνα 3.82 Δυνατότητα πραγματοποίησης κλήσης	84
Εικόνα 3.83 Συνάρτηση StartCall() & onRequestPermissionsResult()	85
Εικόνα 3.84 Επικοινωνία με διατροφολόγο I	86
Εικόνα 3.85 Επικοινωνία με διατροφολόγο II	86
Εικόνα 3.86 Ενδεικτικό πρόγραμμα διατροφής	87
Εικόνα 3.87 Πληροφορίες για άμεση αποκατάσταση.....	88
Εικόνα 3.88 Υπολογισμός θερμιδικών αναγκών.....	88
Εικόνα 3.89 Βασικές πληροφορίες διατροφής.....	89
Εικόνα 3.90 Εμφάνιση και τροποποίηση προπονητικού προγράμματος	90
Εικόνα 3.91 Επίβλεψη δραστηριότητας συγκεκριμένου αθλητή.....	91
Εικόνα 3.92 Εμφάνιση αποτελεσμάτων ψυχολογικού τεστ.....	92
Εικόνα 3.93 Πληροφορίες για τα αποτελέσματα ψυχολογικού τεστ	92
Εικόνα 3.94 Navigation Menu	93
Εικόνα 3.95 Navigation Menu - κώδικας I	93
Εικόνα 3.96 Navigation Menu - κώδικας II	93
Εικόνα 3.97 Security Rules	94

Αλφαβητικό Ευρετήριο

BFI: Big Five Inventory

BMR: Basal Metabolic Rate

HR max: Maximum Heart Rate

VO2 max: Maximum Oxygen Intake

ΕΙΣΑΓΩΓΗ

Η μέγιστη απόδοση αθλητή προϋποθέτει τον καλύτερο δυνατό συνδυασμό των επιστημών της Φυσικής Αγωγής, Διατροφολογίας, Ψυχολογίας και Φυσικοθεραπείας. Το πρόβλημα γίνεται ακόμα πιο περίπλοκο όταν εμφανίζονται εξωτερικοί παράγοντες, που δεν επιτρέπουν την ομαλή εκτέλεση ενός προκαθορισμένου προγράμματος και απαιτούνται διαρκής αλλαγές.

Οι περισσότεροι αθλητές (συμπεριλαμβανομένου αθλητών υψηλού επιπέδου), για να μπορούν να ανταπεξέλθουν στις βιοποριστικές τους ανάγκες, αναγκάζονται να εργαστούν. Αυτό πολλές φορές δεν τους επιτρέπει να ακολουθήσουν το προπονητικό πρόγραμμα που έχει σχεδιαστεί και αναγκάζονται να το αλλάξουν. Οι εφαρμογές επίβλεψης αθλούμενων εστιάζουν κατά κύριο λόγο στις μετρήσεις από διάφορους αισθητήρες (καρδιακών παλμών, καταναλισκόμενης ισχύς, υψομέτρου, βαρομέτρου). Όμως οι πιο διακεκριμένοι προπονητές όμως, εκτός από τις ενδείξεις των αισθητήρων, λαμβάνουν υπόψη τους την αίσθηση που έχει ο κάθε αθλητής και προσαρμόζουν αναλόγως την επιβάρυνση της κάθε προπονητικής μονάδας. Επιπλέον, ιδιαίτερα σημαντικό στην απόδοση του αθλητή είναι η κάλυψη των ενεργειακών του αναγκών μέσω της ισορροπημένης διατροφής.

Σημαντικό ρόλο έχει και ο χαρακτήρας του κάθε αθλητή που απαιτεί διαφορετική προσέγγιση. Με την ένταξη του τεστ για τον προσδιορισμό του ψυχολογικού προφίλ ο προπονητής μπορεί να προσαρμόζει την ημερήσια προπόνηση, αλλά και την συμπεριφορά του απέναντι στον κάθε αθλητή.

Για την καλύτερη δυνατή απόδοση του αθλητή την ημέρα της δοκιμασίας, όλα τα παραπάνω πρέπει συνυπάρχουν αρμονικά καθ' όλη τη διάρκεια της περιόδου προετοιμασίας. Τις περισσότερες φορές οι αθλητές καλούνται να αντιμετωπίσουν όλα τα παραπάνω ολομόναχοι και οι υπάρχουσες εφαρμογές δυσκολεύονται στο να βρουν τον ιδανικό συνδυασμό προπόνησης (αλλά και των αλλαγών του προγράμματος), διατροφής και αποκατάστασης για τον κάθε χρήστη ξεχωριστά.

Αντικείμενο της διπλωματικής εργασίας

Το αντικείμενο αυτής της εργασίας είναι η εφαρμογή σύγχρονων μέσων για την ανάπτυξη εφαρμογής η οποία είναι συμβατή με τα όλα τα μοντέλα κινητών τηλεφώνων που χρησιμοποιούν λογισμικό Android. Η εφαρμογή έχει ως σκοπό την επίβλεψη της (φυσικής, ψυχολογικής κ.α.) κατάστασης του αθλούμενου. Είναι αρκετά ενδιαφέρον θέμα διότι η εφαρμογή εκτός από τις πληροφορίες που παρέχει στον ίδιο τον αθλητή, βοηθάει και τον προπονητή στη λήψη αποφάσεων για τυχόν αλλαγές, όχι μόνο με βάση τα δεδομένα από τους αισθητήρες, άλλα στηρίζεται και στο ψυχολογικό προφίλ του αθλούμενου και στην αίσθησή του.

Σκοπός και στόχοι

Σκοπός της εργασίας ήταν η δημιουργία μιας ολοκληρωμένης εφαρμογής για την επίβλεψη της πορείας του αθλητή.

Πιο συγκεκριμένα οι στόχοι της εργασίας διακρίνονται ως εξής:

- Κατασκευή μιας σύγχρονης και ασφαλούς εφαρμογής για όλα τα κινητά που χρησιμοποιούν λογισμικό Android.
- Η εφαρμογή να παρέχει εύκολο και φιλικό για το χρήστη περιβάλλον.

- Δημιουργία διαφορετικών προπονητικών προγραμμάτων που θα προσαρμόζονται αυτόματα στο επίπεδο του κάθε αθλούμενου.
- Προσαρμογή του προπονητικού προγράμματος ανάλογα με τον δείκτη κόπωσης του αθλούμενου, με στόχο τη μέγιστη αύξηση της απόδοσης, αλλά και την μείωση της πιθανότητας τραυματισμού ή υπερκόπωσης.
- Ένταξη τεστ για τον προσδιορισμό του ψυχολογικό προφίλ του αθλητή που βοηθάει τον προπονητή στην καλύτερη κατανόηση του τρόπου σκέψης του κάθε αθλητή.
- Δημιουργία προγράμματος διατροφής που θα προσαρμόζεται αυτόματα στα σωματομετρικά χαρακτηριστικά του κάθε αθλητή, τις θερμιδικές ανάγκες του, σε συνάρτηση με την προπονητική επιβάρυνση.
- Εξασφάλιση στον προπονητή της δυνατότητας μορφοποίησης του προπονητικού προγράμματος, αλλά και την επιλογή επίβλεψης της πορείας επιλεγμένων αθλητών.
- Εξασφάλιση άμεσης επικοινωνίας μεταξύ προπονητή - αθλούμενου.

Μεθοδολογία

Για την εκπόνηση της διπλωματικής ακολουθήθηκαν τα παρακάτω βήματα.

- Πραγματοποιήθηκε εκτεταμένη έρευνα στο διαδίκτυο, για παρόμοιες εφαρμογές παρακολούθησης αθλητών. Συγκεντρώθηκε η απαιτούμενη βιβλιογραφία η οποία αποτελείται από επιστημονικές δημοσιεύσεις και άρθρα που αφορούν τους τομείς Φυσικής Αγωγής, Εργοφυσιολογίας, Ψυχολογίας και Διατροφής. Επιλέχθηκαν τα κατάλληλα τεχνολογικά εργαλεία που ήταν απαραίτητα για την υλοποίηση της εφαρμογής.
- Μελετήθηκε ο τρόπος κατασκευής εφαρμογών με τη χρήση Android Studio και η γλώσσα προγραμματισμού Kotlin. Ακολούθησε ο σχεδιασμός της βασικής δομής της εφαρμογής, αλλά και όλης της βάσης δεδομένων Realtime Database που παρέχει η πλατφόρμα της Firebase.
- Κατασκευάστηκε η εφαρμογή και υλοποιήθηκε η σύνδεση της με την επιλεγμένη βάση δεδομένων.
- Καταγράφηκαν τα προβλήματα που προέκυψαν κατά την διαδικασία σχεδίασης και υλοποίησης της εφαρμογής. Τέλος, σημειώθηκαν τα συμπεράσματα, οι παρατηρήσεις και παρουσιάστηκαν κάποιες προτάσεις για πιθανή μελλοντική εξέλιξη της εφαρμογής.

Καινοτομία

Στις διαδεδομένες εφαρμογές παρακολούθησης και καθοδήγησης του αθλητή δε λαμβάνεται υπόψη η αίσθηση που έχει ο ίδιος ο αθλούμενος πριν και μετά την άσκηση, αλλά ούτε και το ψυχολογικό του προφίλ. Αρκετές φορές έχει αποδειχθεί ότι ακούγοντας την γνώμη του αθλητή έχουν αποφευχθεί διάφοροι τραυματισμοί. Άλλος ένας σημαντικός παράγοντας που συχνά εμποδίζει την εκτέλεση ενός κάθετου προγράμματος είναι η καθημερινή δραστηριότητα του αθλούμενου, που δε λαμβάνεται υπόψη. Η εφαρμογή που πραγματοποιήθηκε στα πλαίσια αυτής της διπλωματικής ενώνει σε ένα αρκετά μεγάλο βαθμό το κομμάτι του προπονητή, διατροφολόγου και του ψυχολόγου. Είναι ένα εργαλείο που παρέχει στον προπονητή μια ολοκληρωμένη εικόνα της κατάστασης στην οποία βρίσκεται ένας αθλητής και του επιτρέπει να προχωρήσει σε άμεσες αλλαγές.

Δομή

Στο κεφάλαιο 1, αναλύονται οι βασικές έννοιες που σχετίζονται με τις επιστήμες της Φυσικής Αγωγής και Διατροφολογίας. Εξηγείται η σημασία του ψυχολογικού προφίλ, αλλά και η άμεση σύνδεση του με την απόδοση του αθλούμενου. Στη συνέχεια, στην ενότητα 2 αναλύεται το τεχνικό υπόβαθρο και τα μέσα που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Κατόπιν, στην ενότητα 3 παρουσιάζεται η δομή και λειτουργία της εφαρμογής. Αναλύονται τα επιμέρους μέρη και εξηγείται ο τρόπος λειτουργίας τους. Τέλος, στην ενότητα 4, στον επίλογο βρίσκεται η σύνοψη της εργασίας, τα προβλήματα και η αντιμετώπιση τους, οι παρατηρήσεις, τα συμπεράσματα και οι προτάσεις μελλοντικής εξέλιξης της εφαρμογής.

1 ΚΕΦΑΛΑΙΟ 1^ο : Θεωρητικό Υπόβαθρο

1.1 Εισαγωγή κεφαλαίου

Σε αυτό το κεφάλαιο περιγράφονται βασικές έννοιες που σχετίζονται με την επιστήμη της Φυσικής Αγωγής, Διατροφολογίας, Φυσικοθεραπείας και Ψυχολογίας. Εξηγείται επίσης η σύνδεση που υπάρχει ανάμεσα τους στην απόδοση του αθλούμενου.

Στην ενότητα 1.2 αναλύονται οι βασικοί παράγοντες που συντελούν στην απόδοση του αθλητή. Επιπλέον, στην ενότητα 1.3 αναλύονται οι βασικές έννοιες της προπονητικής, τα ενεργειακά συστήματα του ανθρώπου και όλες οι θεωρητικές γνώσεις που απαιτήθηκαν για την υλοποίηση του προπονητικού προγράμματος που ενσωματώθηκε στην εφαρμογή.

Στη συνέχεια περιγράφεται η μέθοδος προσδιορισμού ψυχολογικού προφίλ του αθλούμενου που επιλέχθηκε να ενταχθεί στην εφαρμογή. Τέλος, στην ενότητα 1.5 αυτού του κεφαλαίου περιγράφονται τα συστήματα που χρησιμοποιούνται για την παρακολούθηση της απόδοσης ενός αθλητή.

1.2 Παράγοντες που συντελούν στην απόδοση του αθλητή

1.2.1 Βασικές αρχές της προπονητικής

Σε αυτή την ενότητα περιγράφονται οι βασικές αρχές προπονητικής που κάθε προπονητής οφείλει να ακολουθεί κατά τη δημιουργία προπονητικού προγράμματος.

- **Η αρχή της βελτίωσης της υγείας**

Η αρχή αυτή αναφέρεται στην οργάνωση της προπόνησης με τρόπο που το περιεχόμενο της να έχει αρχικά μια ελάχιστη ένταση, ώστε να προκαλεί στον οργανισμό τη δυνατότητα προσαρμογής και από την άλλη η μέγιστη ένταση να αποκλείει την υπερπροπόνηση, που μπορεί να οδηγήσει σε υπερκόπωση. Η προπονητική επιβάρυνση κατά τη διάρκεια της άσκησης πρέπει να καθορίζεται πάντοτε σύμφωνα με το φύλο και την ηλικία του αθλούμενου. Η προπονητική διαδικασία που τηρεί αυτή την αρχή, εξυπηρετεί το πραγματικό νόημα του αθλητισμού και οδηγεί τον αθλούμενο στην απόκτηση συνηθειών σωστής προσωπικής υγιεινής, τόσο κατά τη διάρκεια της άσκησης όσο και μετά από αυτήν.

- **Η αρχή της ωφελιμότητας**

Τα αθλητικά προγράμματα πρέπει να σχεδιάζονται με βάση την επιστημονική θεμελίωση του περιεχομένου τους. Αυτό σημαίνει ότι η θεωρία και η πράξη θα πρέπει να αποτελούν μια ενότητα. Η αρχή της ωφελιμότητας μαζί με τις άλλες αρχές της προπόνησης εξυπηρετεί αυτή τη σύνθεση, επισημαίνοντας το «κοινό» σημείο που υπάρχει μεταξύ των αγωνισμάτων, των απαιτήσεων του προπονητικού επιπέδου της ηλικίας και των μεθόδων.

- **Η αρχή της ευσυνειδησίας**

Αυτή η αρχή έχει τον σημαντικότερο ρόλο στην αγωγή των αθλητών μέσα στο πνεύμα της πειθαρχημένης προπόνησης. Απαιτεί από τον αθλητή να ρυθμίζει και να ισορροπεί τα αθλητικά του ενδιαφέροντα με τον πρωταθλητισμό και την κοινωνική και οικογενειακή του ζωή. Με αυτόν τον τρόπο η αρχή της ευσυνειδησίας, θα επιτρέψει στον αθλητή να γίνει όχι μόνο ένας κορυφαίος εκτελεστής (συγκεκριμένης δραστηριότητας), αλλά επίσης ένα πρότυπο σπουδαστή και επαγγελματία.

- **Η αρχή της Επανάληψης**

Η αρχή της επανάληψης είναι ιδιαίτερα σημαντική σε ότι αφορά την ανάπτυξη των φυσικών κινητικών προτύπων. Με την προοδευτική αύξηση του αριθμού των επαναλήψεων οι κινήσεις γίνονται αυτόματες, ενώ ταυτόχρονα αναπτύσσονται η ταχύτητα, η δύναμη και η αντοχή. Καθορίζει την ολική επιβάρυνση κατά τη διάρκεια μιας προπονητικής περιόδου, καθώς και τη συχνότητα των προπονητικών περιόδων.

- **Η αρχή της συστηματοποίησης**

Η αρχή αυτή εφαρμόζεται για την εκμάθηση της βασικής τεχνικής αλλά και για τη βελτίωση και τον συντονισμό της δύναμης, της ταχύτητας, της αντοχής, της ευκινήσιμης και της ευλυγισίας. Η συστηματική αύξηση της άσκησης κατά τη διάρκεια μιας προπονητικής περιόδου, πρέπει να ακολουθεί την αρχή της συστηματοποίησης και τα δύο συνθετικά της, που είναι η δυναμικότητα και η συνεχής εξάσκηση. Η σχέση μεταξύ της ολικής επιφόρτισης και της περιόδου αποκατάστασης είναι αναλογική. Όσο αυξάνεται ο όγκος και η ένταση της προπόνησης, τόσο περισσότερο ο οργανισμός έχει την ανάγκη για μεγαλύτερη περίοδο ανάληψης.

- **Η αρχή της ανθεκτικότητας**

Η αρχή της ανθεκτικότητας επιδιώκει μια εποικοδομητική προπόνηση, που να εγγυάται τη σταθερότητα επίκτητων ικανοτήτων, τη συντήρηση και τον συντονισμό της για ένα μεγάλο χρονικό διάστημα χωρίς αλλαγή στην ποιότητα της. Είναι στενά συνδεδεμένη με την αρχή της επανάληψης και της συστηματοποίησης. Οι αποκτούμενες ικανότητες συνήθως χάνονται με την πάροδο του χρόνου και σταθεροποιούνται μόνο όταν επαναλαμβάνονται συχνά σε μικρά χρονικά διαστήματα.

- **Η αρχή της προοδευτικότητας**

Σύμφωνα με την αρχή της προοδευτικότητας, το συνολικό φορτίο πρέπει να αυξάνεται βαθμιαία κατά την προγύμναση ατόμων κάθε ηλικίας και ιδιαίτερα όταν πρόκειται για άτομα μικρής ηλικίας. Είναι ο μόνος τρόπος με τον οποίον μπορεί να αποφευχθεί βλάβη στο καρδιαγγειακό σύστημα ή σε οποιοδήποτε άλλο μέρος του σώματος. Η αρχή της προοδευτικότητας για τη βελτίωση της δύναμης, ταχύτητας και αντοχής είναι στενά συνδεδεμένη με την αρχή της επανάληψης.

- **Η αρχή της εξάρτησης από την ηλικία**

Η αρχή της εξάρτησης από την ηλικία επιβάλλει τη διαφοροποίηση των προπονητικών μεθόδων τόσο στους εφήβους και στους ενήλικες, όσο και στους αρχάριους και τους ειδικευμένους. Για το γενικό συντονισμό των φυσικών ικανοτήτων, η πολύπλευρη προπόνηση και η βαθμιαία αύξηση στα προπονητικά φορτία είναι απαραίτητη. Είναι φανερό ότι υπάρχει στενή σχέση μεταξύ της αρχής της εξάρτησης από την ηλικία, της αρχής της προαγωγής της υγείας και της αρχής της προοδευτικότητας.

1.2.2 Διατροφή & Ενεργειακές απαιτήσεις

Η διατροφή παίζει καθοριστικό ρόλο στην αθλητική απόδοση. Παρέχει όλα τα θρεπτικά συστατικά που απαιτούνται για την υποστήριξη της σωματικής δραστηριότητας και βοηθά το σώμα να ανακάμψει μετά την άσκηση. Είναι σημαντικό για τους αθλητές να έχουν μια ισορροπημένη διατροφή που να καλύπτει τις ανάγκες τους. Η επαρκής διατροφή συμβάλλει στην αύξηση των επιπέδων ενέργειας, στη βελτίωση της αντοχής και στην καθυστέρηση της κόπωσης. Βοηθά επίσης

στην οικοδόμηση και την αποκατάσταση του μυϊκού ιστού, στη διατήρηση ενός υγιούς σωματικού βάρους και στην ενίσχυση της συνολικής υγείας.

Ειδικότερα, οι υδατάνθρακες και η ενυδάτωση είναι σημαντικές για την αθλητική απόδοση. Οι υδατάνθρακες αποτελούν την κύρια πηγή ενέργειας για το σώμα κατά τη διάρκεια της σωματικής δραστηριότητας και μπορούν να βοηθήσουν στη βελτίωση της αθλητικής απόδοσης. Σε κάποια αθλήματα συνίσταται και η λήψη υδατανθράκων κατά τη διάρκεια της προπόνησης ή του αγώνα. Η διατήρηση επαρκής ενυδάτωσης είναι εξαιρετικά σημαντική, καθώς ακόμη και η ήπια αφυδάτωση μπορεί να επηρεάσει αρνητικά την απόδοση.

Επιπλέον, είναι σημαντικό για τους αθλητές να καταναλώνουν αρκετή πρωτεΐνη για την υποστήριξη της μυϊκής ανάπτυξης αλλά και της επίσπευση της διαδικασίας μυϊκής αποκατάστασης μετά από μια δύσκολη προπονητική μονάδα. Η επαρκής πρόσληψη βιταμινών και μετάλλων, όπως ο σίδηρος, το ασβέστιο οι βιταμίνες C, D, αλλά και του συμπλέγματος βιταμίνης B, είναι επίσης απαραίτητη για τη βέλτιστη υγεία και την αθλητική απόδοση. Η διατροφή είναι άμεσα συνδεδεμένη με το προπονητικό πρόγραμμα που εκτελεί ο κάθε αθλητής. Οι ενεργειακές απαιτήσεις διαφέρουν ανάλογα με τα σωματομετρικά χαρακτηριστικά (ύψος, βάρος, μυϊκή μάζα), την ηλικία, το φύλλο, το επίπεδο ημερήσιας δραστηριότητας, αλλά και το προπονητικό πρόγραμμα που ακολουθεί ο αθλούμενος.

Ο βασικός μεταβολισμός (BMR- Basal Metabolic Rate) αντιπροσωπεύει τις ενεργειακές απαιτήσεις που έχει ο οργανισμός, για μπορεί να ανταπεξέλθει στις βασικές λειτουργίες κυττάρων και ιστών σε κατάσταση ηρεμίας. Υπολογίζεται λαμβάνοντας υπόψη το ύψος, το βάρος, την ηλικία και το φύλλο του εξεταζόμενου [38].

Υπάρχουν αρκετές διαφορετικές μέθοδοι υπολογισμού του βασικού μεταβολισμού .

Στην εφαρμογή που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας επιλέχθηκε η **μέθοδος Harris-Benedict** καθώς διακρίνεται από ιδιαίτερη ακρίβεια [38].

Ο τύπος για τον υπολογισμό του BMR για άντρες:

$$\text{BMR} = 66 + (13,7 * \text{weight [kg]}) + (5 * \text{height [cm]}) - (6,8 * \text{age [years]})$$

Ο αντίστοιχος τύπος υπολογισμού για τις γυναίκες:

$$\text{BMR} = 655 + (9,57 * \text{weight [kg]}) + (1,85 * \text{height [cm]}) - (4,68 * \text{age [years]})$$

Υπάρχουν κάποιες ειδικές κατηγορίες ατόμων, όπου ο δείκτης αυτός είναι διαφορετικός και δεν μπορεί να προσεγγιστεί με ακρίβεια. Σε αυτές ανήκουν τα άτομα νεαρής ηλικίας (13-19 ετών) αλλά και οι έγκυες γυναίκες.

Το επίπεδο της σωματικής δραστηριότητας έχει σημαντικό ρόλο στον καθορισμό των ημερήσιων ενεργειακών αναγκών, γι' αυτό και αποτελεί σημαντικό παράγοντα που πρέπει να λαμβάνετε υπόψη όταν σχεδιάζετε η διατροφή του αθλούμενου.

Η τιμή του δείκτη σωματικής δραστηριότητας κυμαίνεται από 1,2 μέχρι 2,0

- **Καθιστός: BMR x 1,2**

Αυτό το επίπεδο δραστηριότητας απευθύνεται σε άτομα που έχουν καθιστική ζωή, άτομα που περνούν το μεγαλύτερο μέρος της ημέρας τους καθιστοί ή ξαπλωμένοι.

- **Ελαφρώς δραστήριος: BMR x 1,4**

Αυτό το επίπεδο δραστηριότητας απευθύνεται σε άτομα που έχουν αρκετά καθιστική ζωή, αλλά υπάρχουν στιγμές τις ημέρας που απαιτείται μια μικρή σωματική δραστηριότητα.

- **Μέτρια ενεργός: BMR x 1,6**

Αυτό το επίπεδο δραστηριότητας είναι για άτομα που έχουν μέτριο επίπεδο

δραστηριότητας, όπως εκείνοι που ασκούν μέτρια σωματική δραστηριότητα για 30-60 λεπτά την ημέρα.

- **Πολύ δραστήριος: BMR x 1,8**

Αυτό το επίπεδο δραστηριότητας είναι για άτομα που έχουν υψηλό επίπεδο δραστηριότητας, όπως εκείνοι που ασκούν έντονη σωματική δραστηριότητα για 60 λεπτά ή περισσότερο την ημέρα ή έχουν μια σωματικά απαιτητική εργασία.

- **Εξαιρετικά δραστήριος: BMR x 2,0**

Αυτό το επίπεδο δραστηριότητας είναι για άτομα που έχουν πολύ υψηλό επίπεδο δραστηριότητας, όπως οι αθλητές αγωνιστικού επιπέδου που ασχολούνται με έντονη σωματική δραστηριότητα για αρκετές ώρες την ημέρα.

Είναι σημαντικό να σημειωθεί ότι αυτοί οι συντελεστές δραστηριότητας είναι απλώς εκτιμήσεις και μπορεί να μην είναι απόλυτα ακριβείς. Υπάρχουν και άλλοι παράγοντες, όπως η σύνθεση του σώματος, η γενετική και η ενεργειακή πρόσληψη που μπορούν επίσης να επηρεάσουν τις ημερήσιες ενεργειακές ανάγκες. Για τον ακριβή υπολογισμό των ημερήσιων ενεργειακών αναγκών με τη μέγιστη δυνατή ακρίβεια, απαιτείται η παρακολούθηση του αθλούμενου από έναν διατροφολόγο.

1.2.3 Σημασία της ξεκούραση - αποκατάστασης

Η ανάπαυση και η αποκατάσταση είναι κρίσιμα συστατικά κάθε προγράμματος προπόνησης στον αθλητισμό. Αφιερώνοντας χρόνο για ξεκούραση και αποκατάσταση, οι αθλητές μπορούν να μειώσουν τον κίνδυνο τραυματισμού, να βελτιώσουν την απόδοση, να ενισχύσουν την ψυχική ευεξία και να επισπεύσουν τη μυϊκή ανάπτυξη.

Ακολουθούν ορισμένοι λόγοι για τους οποίους η σωστή ξεκούραση και αποκατάσταση είναι εξίσου σημαντικά με την προπόνηση:

- **Μειώνεται ο κίνδυνος τραυματισμού:** Η υπερβολική καταπόνηση του οργανισμού και η εξάντληση είναι κοινές αιτίες τραυματισμών στους αθλητές. Η λήψη επαρκούς χρόνου ανάπαυσης και αποκατάστασης μπορεί να συμβάλει στη μείωση του κινδύνου τραυματισμού, επιτρέποντας στο σώμα να επιδιορθωθεί και αποφεύγοντας την υπερπροσπάθεια.
- **Βελτιώνεται η απόδοση:** Το σώμα χρειάζεται χρόνο για να ανακάμψει και να αναδομηθεί μετά από έντονη άσκηση. Η πραγματοποίηση τακτικών διαλειμμάτων και η ενσωμάτωση ημερών ξεκούρασης στο πρόγραμμα προπόνησης μπορεί να βελτιώσει αισθητά την απόδοση, αποτρέποντας την εξάντληση και επιτρέποντας στο σώμα να αποδίδει τα μέγιστα.
- **Βελτιώνεται η ψυχική ευεξία:** Η προπόνηση μπορεί να είναι τόσο ψυχικά όσο και σωματικά απαιτητική. Η αφιέρωση χρόνου για ξεκούραση και αποκατάσταση μπορεί να βοηθήσει στη βελτίωση της διάθεσης, στη μείωση του άγχους και στην πρόληψη της ψυχολογικής εξουθένωσης.
- **Βοηθάει την ανάπτυξη των μυών:** Κατά τη διάρκεια της ανάπαυσης, το σώμα επιδιορθώνει και αναδομεί τον μυϊκό ιστό. Η επαρκής ανάπαυση είναι απαραίτητη για τη μυϊκή ανάπτυξη και εξέλιξη, η οποία είναι απαραίτητη για τη βελτίωση της αθλητικής απόδοσης.

1.2.4 Ψυχολογικός παράγοντας

Ο ψυχολογικός παράγοντας έχει κρίσιμο ρόλο στην απόδοση του αθλητή στον αγώνα αλλά και στην προπόνηση. Οι αθλητές μπορούν να επωφεληθούν από την ενσωμάτωση ασκήσεων για τη βελτίωση διάφορων στοιχείων του χαρακτήρα τους. Ασκήσεις που αυξάνουν την αυτοπεποίθηση αλλά και το επίπεδο ψυχικής ανθεκτικότητας. Με τη σωστή διαχείριση του ψυχολογικού παράγοντα ο καθορισμός των στόχων και η συνεχής προσήλωση σε αυτούς γίνεται όλο και πιο εύκολα με την πάροδο του χρόνου.

1.2.5 Εξωτερικοί παράγοντες

Αρκετές φορές το επίπεδο απόδοσης του αθλητή την ημέρα της δοκιμασίας εξαρτάται άμεσα από εξωτερικούς παράγοντες, τους οποίους δεν μπορεί να ελέγξει και οφείλει να προσαρμοστεί σε αυτούς με τον καλύτερο δυνατό τρόπο. Ιδιαίτερα στα αγωνίσματα μεσαίων και μεγάλων αποστάσεων, όπου ο αγωνιστικός χώρος στις περισσότερες περιπτώσεις είναι σε εξωτερικό χώρο, η σωστή προσαρμογή παίζει καθοριστικό ρόλο.

Καιρικές συνθήκες

- **Θερμοκρασία**
Είναι επιστημονικά αποδεδειγμένο ότι οι πολύ υψηλές αλλά και οι πολύ χαμηλές θερμοκρασίες έχουν αρνητική επίπτωση στην απόδοση του αθλούμενου. Σε πολύ χαμηλές θερμοκρασίες ο οργανισμός δυσκολεύεται να διατηρήσει τη σταθερή θερμοκρασία του σώματος, με αποτέλεσμα οι ενεργειακά απαιτήσεις να αυξάνονται. Μια ιδιαίτερα απαιτητική σωματική προσπάθεια σε τέτοιες δύσκολες συνθήκες απαιτεί μια μεγαλύτερη περίοδο αποκατάστασης.
- **Υγρασία**
Στα αγωνίσματα δρόμων αντοχής, όταν το επίπεδο υγρασίας υπερβαίνει το 60%, έχει αποδειχθεί ότι ο αθλούμενος μπορεί να αντιμετωπίσει δυσκολίες στην αναπνοή και άμεση συνέπεια αυτού είναι η αύξηση των καρδιακών του παλμών.

Πολλές φορές ο συνδυασμός των παραπάνω μπορεί να είναι αυτός που θα κρίνει την επιτυχία ή την αποτυχία του αθλητή την ημέρα του αγώνα.

Τεχνολογικός παράγοντας

- **Ποιότητα αγωνιστικού χώρου:** Στα δρομικά αγωνίσματα το είδος και η ποιότητα του ταρτάν ή πρόσφυση της ασφάλτου.
- **Ενδυμασία/ Υπόδηση**
Τα τελευταία χρόνια, ο τομέας κατασκευής αθλητικής ενδυμασίας και υπόδησης σημειώνει μεγάλη πρόοδο. Η αθλητική ενδυμασία παράγεται από ελαφριά υλικά, που επιτρέπουν στο δέρμα να αερίζεται επαρκώς, ο οργανισμός δεν υπερθερμαίνεται, πράγμα το οποίο συμβάλει στη μικρότερη ενεργειακή σπατάλη.
Η ενσωμάτωση των carbon plates στα αγωνιστικά παπούτσια έφερε την επανάσταση στον δρομικό κόσμο. Έχει αποδειχθεί ότι συγκεκριμένα μοντέλα βελτιώνουν την απόδοση του αθλητή κατά μέσο όρο 4-5%, μειώνοντας συγχρόνως την καταπόνηση του οργανισμού το οποίο συνδέεται άμεσα με μικρότερη περίοδο αποκατάστασης που ακολουθεί μετά από έναν απαιτητικό αγώνα.

- Εργαλεία παρακολούθησης δραστηριότητας.
Επιτρέπουν τη συλλογή δεδομένων μέσω αισθητήρων (στον καρπό ή μέσω ζώνης που τοποθετείται στο στήθος) και παρέχουν συμβουλές σε πραγματικό χρόνο. Τα σύγχρονα ρολόγια βοηθάνε στην παρακολούθηση διάφορων παραμέτρων, όπως καρδιακοί παλμοί, καταναλισκόμενη ισχύς, με σκοπό τη βοήθεια στη διατήρηση σωστού ρυθμού.

1.3 Βασικές έννοιες στην προπονητική

Σε αυτή ενότητα περιγράφονται βασικές έννοιες που συνδέουν τη φυσιολογία του ανθρώπου με τον τομέα της προπονητικής.

1.3.1 Ενεργειακά συστήματα του ανθρώπου

Το ανθρώπινο σώμα αποθηκεύει ενέργεια με διάφορους τρόπους. Για να χρησιμοποιηθεί αυτή η ενέργεια για την παραγωγή μυϊκών συστολών και κίνησης, πρέπει να γίνουν συγκεκριμένες βιοχημικές αντιδράσεις στο μυ. Αυτές οι βιοχημικές αντιδράσεις λειτουργούν ως βάση για την ταξινόμηση της ανθρώπινης ενεργειακής δαπάνης μέσω τριών ενεργειακών συστημάτων: το σύστημα ATP-PC, το σύστημα γαλακτικού οξέος και το σύστημα οξυγόνου.

- **Σύστημα ATP-PC**

Είναι γνωστό και ως σύστημα των φωσφογόνων, επειδή τόσο το ATP όσο και η PC περιέχουν φωσφορικό οξύ. Το ATP είναι η άμεση πηγή ενέργειας για όλες σχεδόν τις σωματικές διεργασίες, συμπεριλαμβανομένης και της μυϊκής συστολής. Η PC είναι μια ένωση υψηλής ενέργειας που βρίσκεται στους μύες και μπορεί να βοηθήσει ταχύτατα τη σύνθεση ATP καθώς το τελευταίο χρησιμοποιείται. Το σύστημα ATP-PC δραστηριοποιείται σε μυϊκές προσπάθειες υψηλής ισχύος αλλά μικρής διάρκειας. Τα ATP και PC βρίσκονται σε μικρές ποσότητες και κάθε έντονη άσκηση διάρκειας 5-10 δευτερολέπτων θα μπορούσε να εξαντλήσει την ποσότητα τους σε οποιαδήποτε μυ. Για αυτό το λόγο, πρέπει να αναπληρώνονται άμεσα και αυτή είναι η λειτουργία των άλλων ενεργειακών συστημάτων [38].

- **Σύστημα γαλακτικού οξέος (Anaerobic Metabolism)**

Δεν μπορεί να χρησιμοποιηθεί άμεσα ως πηγή ενέργειας για τη μυϊκή συστολή, αλλά μπορεί να βοηθήσει στη γρήγορη ανασύνθεση του ATP όταν είναι απαραίτητο. Για την άσκηση υψηλής έντασης η αμέσως επόμενη καλύτερη ενεργειακή πηγή μετά την ATP-PC είναι το μυϊκό γλυκογόνο. Το μυϊκό γλυκογόνο, για να χρησιμοποιηθεί ως ενέργεια, πρέπει να διασπαστεί σε γλυκόζη, η οποία υφίσταται μια σειρά αντιδράσεων για να σχηματιστεί τελικά ATP, μια διαδικασία που ονομάζεται γλυκόλυση. Παράγει περιορισμένη ποσότητα ATP και τη μέγιστη γαλακτικού. Κατά τον αναερόβιο μεταβολισμό, η ενέργεια παράγεται χωρίς τη χρήση οξυγόνου και αποθηκεύεται με τη μορφή ATP και φωσφορικής κρεατίνης. Δραστηριοποιείται σε μυϊκές προσπάθειες παρατεταμένης ταχύτητας, μέσης διάρκειας. Ωστόσο, αυτή η πηγή ενέργειας είναι περιορισμένη και μπορεί να διατηρηθεί μόνο για μικρά χρονικά διαστήματα έως 2 λεπτά [38].

- **Σύστημα οξυγόνου (Aerobic Metabolism)**

Κατά τη διάρκεια του αερόβιου μεταβολισμού, το οξυγόνο χρησιμοποιείται για τη διάσπαση της γλυκόζης και των λιπαρών οξέων παρουσία του ενζύμου οξυγόνο (ATP-CP). Παράγει απεριόριστη ποσότητα ATP, την κύρια πηγή ενέργειας του οργανισμού, η οποία τροφοδοτεί τους εργαζόμενους μύες και απειροελάχιστη ποσότητα γαλακτικού.

Σημαντικά χαρακτηριστικά των ανθρώπινων ενεργειακών συστημάτων *				
	ATP - PC	Γαλακτικού οξέος	Οξυγόνου	Οξυγόνου
Βασική πηγή ενέργειας	ATP, φωσφοκρεατίνη	Υδατάνθρακες	Υδατάνθρακες	Λίπη
Επίπεδο έντασης	Ανώτατο	Υψηλό	Χαμηλό	Χαμηλότατο
Ρυθμός παραγωγής ATP	Ανώτατος	Υψηλός	Χαμηλός	Χαμηλότατος
Παραγωγή ισχύος	Ανώτατη	Υψηλή	Χαμηλή	Χαμηλότατη
Δυνατότητα για συνολική παραγωγή ATP	Χαμηλότατη	Χαμηλή	Υψηλή	Ανώτατη
Δυνατότητα αντοχής	Χαμηλότατη	Χαμηλή	Υψηλή	Ανώτατη
Ανάγκη παρουσίας οξυγόνου	Όχι	Όχι	Ναι	Ναι
Αναερόβιο/αερόβιο	Αναερόβιο	Αναερόβιο	Αερόβιο	Αερόβιο
Χαρακτηριστικό αγώνισμα στίβου	100 μέτρα ταχύτητας	400-800 μέτρα	5.000 μέτρα	Υπεραντοχής
Χρονική διάρκεια	1-10 δευτερόλεπτα	10-120 δευτερόλεπτα	5 λεπτά ή περισσότερο	Ώρες
Κατά τη διάρκεια των περισσότερων ασκήσεων, θα λειτουργούν και τα τρία ενεργειακά συστήματα σε κάποιο βαθμό. Παρ' όλα αυτά, μπορεί να επικρατεί ένα σύστημα ανάλογα με την ένταση της άσκησης.				

Πίνακας 1.1 Ανθρώπινα ενεργειακά συστήματα [38]

1.3.2 Αερόβια Αντοχή

Αναφέρεται στη γενική ικανότητα του ατόμου να αντεπεξέρχεται στην κόπωση για μεγάλα χρονικά διαστήματα με την παρουσία όμως σημαντικής προμήθειας οξυγόνου. Στην προπονητική ορολογία υπονοείται η ικανότητα αντίστασης κατά της κόπωσης κάτω από συνθήκες όπου το προσλαμβανόμενο οξυγόνο αλλά και οι οξυγονικές ανάγκες της δραστηριότητας διατηρούνται σταθερά και στο ίδιο επίπεδο. Αυτό το χαρακτηριστικό αναφέρεται ως καρδιαγγειακή ή περιφερειακή αναπνευστική αντοχή και εκδηλώνεται περισσότερο σε έργο μέτριας έντασης και με τη συμμετοχή όλων των λειτουργιών του οργανισμού.

1.3.3 Αερόβιο κατώφλι

Για τους δρομείς, το αερόβιο κατώφλι είναι μια σημαντική έννοια, καθώς αντιπροσωπεύει το σημείο στο οποίο το σώμα μεταβαίνει από τη χρήση κυρίως του λίπους ως καύσιμο στη χρήση υδατανθράκων. Το αερόβιο κατώφλι για τους δρομείς μπορεί να προσδιοριστεί με τη χρήση διαφόρων μεθόδων, συμπεριλαμβανομένης της παρακολούθησης του καρδιακού ρυθμού, της ανάλυσης αερίων και των δοκιμασιών υπό μέγιστης άσκησης [27]. Μόλις προσδιοριστεί το αερόβιο κατώφλι, οι δρομείς μπορούν να το χρησιμοποιήσουν για να σχεδιάσουν τα προπονητικά τους προγράμματα. Για παράδειγμα, μπορούν να προπονηθούν στο αερόβιο κατώφλι ή κοντά σε αυτό για να βελτιώσουν την ικανότητά τους να χρησιμοποιούν το λίπος ως πηγή καυσίμου, οδηγώντας σε βελτιωμένη αντοχή και βραδύτερη εμφάνιση κόπωσης κατά την άσκηση.

1.3.4 Αναερόβια Αντοχή

Είναι η ικανότητα ανοχής στην κόπωση όλων των λειτουργιών του οργανισμού, όταν η παροχή οξυγόνου είναι ανεπαρκής. Στα δρομικά αγωνίσματα αυτή η μορφή αντοχής αποτελεί σημαντικό παράγοντα των αγωνισμάτων 400μ έως 1500μ, αλλά και υπολογίσιμο παράγοντα στα αγωνίσματα μεγάλων αποστάσεων. Αναερόβια αντοχή μπορεί να ονομαστεί και η ταχύτητα αντοχής.

1.3.5 Αναερόβιο κατώφλι (Threshold)

Το αναερόβιο κατώφλι είναι στο σημείο στο οποίο το σώμα μεταβαίνει από τον αερόβιο μεταβολισμό στον αναερόβιο μεταβολισμό. Ο αερόβιος μεταβολισμός χρησιμοποιεί οξυγόνο για την παραγωγή ενέργειας, σε αντίθεση με τον αναερόβιο. Όταν ο αθλητής εκτελεί μια δραστηριότητα σε υψηλή ένταση, το σώμα δεν μπορεί να λάβει αρκετό οξυγόνο για να καλύψει τις ενεργειακές του απαιτήσεις, οπότε μεταβαίνει στον αναερόβιο μεταβολισμό.

Στους δρόμους μεσαίων και μεγάλων αποστάσεων, το γαλακτικό κατώφλι είναι ένας όρος που χρησιμοποιείται συχνά και αναφέρεται στο σημείο στο οποίο το γαλακτικό οξύ αρχίζει να συσσωρεύεται στην κυκλοφορία του αίματός ταχύτερα από ότι μπορεί να απομακρυνθεί. Αυτή η συσσώρευση γαλακτικού οδηγεί σε έντονη μυϊκή κόπωση.

Η προπόνηση στο κατώφλι ή κοντά στο κατώφλι, μπορεί να βοηθήσει στη βελτίωση της αντοχής και της απόδοσης, καθώς βοηθά το σώμα να γίνει πιο αποτελεσματικό στη χρήση του οξυγόνου και στην αερόβια παραγωγή ενέργειας. Σε προπονητικά προγράμματα που οι εντάσεις διαμορφώνονται με βάση τις ζώνες καρδιακών παλμών του αθλητή, το κατώφλι χρησιμοποιείται συχνά ως σημείο αναφοράς για τον προσδιορισμό της έντασης των επιμέρους προπονητικών μονάδων.

1.3.6 Μέγιστη πρόσληψη οξυγόνου (VO₂_max)

Μέγιστη πρόσληψη οξυγόνου (VO₂ max) είναι το μέτρο της μέγιστης ποσότητας οξυγόνου που μπορεί να χρησιμοποιήσει ένα άτομο κατά τη διάρκεια έντονης άσκησης. Είναι ο ανώτατος όγκος οξυγόνου που καταναλώνουν οι ιστοί ανά λεπτό κατά την έντονη μυϊκή προσπάθεια. Θεωρείται ευρέως ως ένας από τους καλύτερους δείκτες του επιπέδου καρδιαγγειακής ικανότητας και της αερόβιας αντοχής ενός ατόμου. Μπορεί να μετρηθεί μέσω διαφόρων εργαστηριακών εξετάσεων, που συνήθως περιλαμβάνουν μια δοκιμασία διαβαθμισμένης άσκησης σε διάδρομο ή ποδήλατο. Τα αποτελέσματα εκφράζονται συνήθως σε χιλιοστόλιτρα οξυγόνου ανά κιλό σωματικού βάρους ανά λεπτό (ml/kg/min).

Ο προσδιορισμός του VO₂ max μπορεί να γίνει με διάφορους τρόπους [39]

Ένας από αυτούς είναι ο παρακάτω:

$$\text{VO2max} = 15.3 \times (\text{MHR} / \text{RHR})$$

Όπου:

· **MHR** (Maximum heart rate)

$$\text{Μέγιστη καρδιακή συχνότητα (παλμοί/λεπτό)} = 220 - \text{ηλικία}$$

- **RHR** (Resting heart rate)

Παλμοί σε κατάσταση ηρεμίας = Καρδιακοί παλμοί σε 20 δευτερόλεπτα x 3
(παλμοί/λεπτό)

Ένας άλλος τρόπος προσέγγισης της μέγιστης πρόσληψης οξυγόνου είναι με βάση τη δοκιμασία 1.5 μιλίου [40].

$$\text{VO2max} = 483/T + 3.5$$

Όπου

T : ο χρόνος που απαιτήθηκε για την ολοκλήρωση της δοκιμασίας (σε λεπτά)

1.3.7 Ζώνες καρδιακών παλμών (Heart Rate)

Οι ζώνες καρδιακής συχνότητας για τους δρομείς υπολογίζονται συνήθως με βάση τη μέγιστη καρδιακή συχνότητα (HRmax) και χρησιμοποιούνται για τον προσδιορισμό της έντασης της άσκησης. Υπάρχουν διάφοροι τρόποι υπολογισμού της HRmax. Ο πιο αξιόπιστος τρόπος προσδιορισμού της μέγιστης καρδιακής συχνότητας είναι το εργομετρικό τεστ. Ένας ευρέως χρησιμοποιούμενος τύπος είναι 220 μείον την ηλικία του ατόμου.

Υπάρχουν 5 ζώνες καρδιακών παλμών. Σε κάθε ζώνη ο οργανισμός έχει διαφορετική κατανάλωση ενέργειας. Η διάρκεια προπόνησης σε κάθε ζώνη συνδέεται άμεσα με τον χρόνο αποκατάστασης που απαιτείται μετά από μια προπόνηση.

- **Ζώνη 1 (50% με 60% HR max):** Ζώνη προθέρμανσης/ανάκαμψης. Είναι η χαμηλότερη ζώνη έντασης που χρησιμοποιείται για προθέρμανση και αποκατάσταση μετά από μια δύσκολη προπόνηση.
- **Ζώνη 2 (60% με 70% HR max):** Αυτή είναι η ζώνη που χρησιμοποιείται για μεγάλα, αργά τρεξίματα με στόχο τη βελτίωση της βασικής αντοχής. Κατά την άσκηση σε αυτή τη ζώνη επιτυγχάνεται μεγαλύτερη καύση λίπους.
- **Ζώνη 3 Tempo (70% με 80% HR max):** Σε αυτή τη ζώνη εκτελούνται προπονήσεις μεσαίας έντασης, όπως οι ανηφόρες ή συνεχόμενο τρέξιμο με σταθερή ταχύτητα. Προπονήσεις σε αυτή τη ζώνη βοηθάνε στην αύξηση αερόβιας αντοχής.
- **Ζώνη 4 Threshold (80% με 90% HR max):** Βοηθάει στη βελτίωση της αναερόβιας αντοχής. Χρησιμοποιείται για τις δύσκολες προπονήσεις που σκοπεύουν στη βελτίωση της ανοχής στο γαλακτικό οξύ, αλλά και στην εξοικείωση του αθλούμενου με τον ρυθμό του αγώνα.
- **Ζώνη 5 (90% με 100% HR max):** Η ζώνη αυτή χρησιμοποιείται σε διαλειμματικές προπονήσεις μέγιστης έντασης, με στόχο τη βελτίωση της μέγιστης ταχύτητας. Η προπόνηση σε αυτή τη ζώνη βοηθάει επίσης στην αύξηση της μέγιστης πρόσληψης

οξυγόνου (VO₂ max). Η μέγιστη διάρκεια του κάθε κομματιού δεν πρέπει να ξεπερνάει τα 2 λεπτά.

Για τους δρομείς μεσαίων και μεγάλων αποστάσεων είναι σημαντικό στο προπονητικό πρόγραμμα να περιλαμβάνονται προπονήσεις σε όλες τις ζώνες.

1.4 Προσδιορισμός ψυχολογικού προφίλ του αθλητή

1.4.1 Σημασία του ψυχολογικού προφίλ και η σχέση του με την απόδοση του αθλητή

Το ψυχολογικό προφίλ ενός ατόμου έχει σημαντικό ρόλο στην επιτυχία του στον αθλητισμό. Η ψυχική ανθεκτικότητα, τα κίνητρα, η επιμονή και προσήλωση στον στόχο παρά τις αποτυχίες και η αυτοπεποίθηση είναι σημαντικοί ψυχολογικοί παράγοντες που μπορούν να επηρεάσουν την αθλητική απόδοση. Για παράδειγμα, ένας αθλητής που έχει ισχυρή αυτοπεποίθηση και αυτοπειθαρχία είναι πιο πιθανό να αποδώσει καλά υπό πίεση, σε σχέση με ένα άτομο με μειωμένη αυτοπεποίθηση. Ένα άτομο που δεν έχει αυτοπειθαρχία ή δεν είναι ψυχικά ανθεκτικό μπορεί να δυσκολευτεί να φτάσει τον εαυτό του στα όρια ή να διατηρήσει το πρόγραμμα προπόνησης του όταν εμφανιστούν κάποιες δυσκολίες. Η κατανόηση του ψυχολογικού προφίλ ενός αθλητή μπορεί να βοηθήσει στην πλήρη αξιοποίηση των δυνατοτήτων του. Είναι πολύ χρήσιμη πληροφορία και για τους προπονητές, τους οποίους βοηθάει ιδιαίτερα στην ανάπτυξη εξατομικευμένων προγραμμάτων προπόνησης και στη λήψη τεκμηριωμένων αποφάσεων σχετικά με τις τυχόν αλλαγές που πρέπει να γίνουν στο πρόγραμμα προπόνησης.

1.4.2 Big Five Inventory

Το Big Five Inventory (BFI) είναι ένα ευρέως χρησιμοποιούμενο εργαλείο αξιολόγησης της προσωπικότητας που μετρά τα χαρακτηριστικά ενός ατόμου σε πέντε ευρείες διαστάσεις της προσωπικότητας: Ανοιχτότητα (Openness), Ευσυνειδησία (Conscientiousness), Εξωστρέφεια (Extraversion), Συμφωνησιμότητα (Agreeableness) και Νευρωτισμός (Neuroticism). Αυτές οι διαστάσεις βρέθηκαν να είναι οι πιο σημαντικές για τη διευκρίνιση των ατομικών διαφορών στην προσωπικότητα και θεωρούνται σταθερές σε όλους τους πολιτισμούς και με την πάροδο του χρόνου. [21]

Ανοιχτότητα (Openness): Αυτή η διάσταση αναφέρεται στη φαντασία, τη δημιουργικότητα και την εκτίμηση της τέχνης και της ομορφιάς. Οι άνθρωποι που σημειώνουν υψηλή βαθμολογία σε αυτή τη διάσταση είναι περίεργοι, περιπετειώδεις και ανοιχτόμυαλοι, ενώ όσοι σημειώνουν χαμηλή βαθμολογία, είναι πιο παραδοσιακοί και συντηρητικοί.

Ευσυνειδησία (Conscientiousness): Αναφέρεται στο επίπεδο οργάνωσης, υπευθυνότητας και αξιοπιστίας ενός ατόμου. Οι άνθρωποι που σημειώνουν υψηλή βαθμολογία στην Ευσυνειδησία είναι αξιόπιστοι και υπεύθυνοι, ενώ όσοι σημειώνουν χαμηλή βαθμολογία σε αυτή τη διάσταση τείνουν να είναι πιο παρορμητικοί και ανοργάνωτοι.

Εξωστρέφεια (Extraversion): Αυτή η διάσταση αναφέρεται στο επίπεδο εξωστρέφειας, ομιλητικότητας και κοινωνικότητας ενός ατόμου. Τα άτομα που σημειώνουν υψηλή βαθμολογία στην Εξωστρέφεια είναι ενεργητικά και απολαμβάνουν να βρίσκονται κοντά σε άλλους, ενώ εκείνοι που σημειώνουν χαμηλή βαθμολογία σε αυτή τη διάσταση είναι πιο συγκρατημένοι και προτιμούν να είναι μόνοι τους.

Συμφωνησιμότητα (Agreeableness): Αυτή η διάσταση αναφέρεται στο επίπεδο συμπόνιας, συνεργασίας και συμπάθειας ενός ατόμου. Τα άτομα που σημειώνουν υψηλή βαθμολογία σε αυτή τη διάσταση διακρίνονται από μεγαλύτερη εν συναίσθηση και είναι πιο εύκολο να συνεργαστεί κάποιος μαζί τους. Εκείνοι που σημειώνουν χαμηλή βαθμολογία σε αυτή τη διάσταση είναι πιο επικριτικοί και αντιδραστικοί.

Νευρωτισμός (Neuroticism): Αυτή η διάσταση αναφέρεται στο επίπεδο συναισθηματικής αστάθειας και άγχους ενός ατόμου. Τα άτομα που σημειώνουν υψηλή βαθμολογία στον νευρωτισμό, τείνουν να είναι πιο ευαίσθητα, αγχώδη και επιρρεπή στην ανησυχία, ενώ εκείνοι που σημειώνουν χαμηλή βαθμολογία, είναι πιο σταθεροί συναισθηματικά και ανθεκτικοί.

Big Five Dimensions	Facet (and correlated trait adjective)
Extraversion vs. introversion	Gregariousness (sociable) Assertiveness (forceful) Activity (energetic) Excitement-seeking (adventurous) Positive emotions (enthusiastic) Warmth (outgoing)
Agreeableness vs. antagonism	Trust (forgiving) Straightforwardness (not demanding) Altruism (warm) Compliance (not stubborn) Modesty (not show-off) Tender-mindedness (sympathetic)
Conscientiousness vs. lack of direction	Competence (efficient) Order (organized) Dutifulness (not careless) Achievement striving (thorough) Self-discipline (not lazy) Deliberation (not impulsive)
Neuroticism vs. emotional stability	Anxiety (tense) Angry hostility (irritable) Depression (not contented) Self-consciousness (shy) Impulsiveness (moody) Vulnerability (not self-confident)
Openness vs. closedness to experience	Ideas (curious) Fantasy (imaginative) Aesthetics (artistic) Actions (wide interests) Feelings (excitable) Values (unconventional)

Εικόνα 1.1 Οι πέντε μεγάλοι παράγοντες (Big Five Factors)

Στην εφαρμογή που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής, εφαρμόστηκε το τεστ αυτοαξιολόγησης των 44 ερωτήσεων (International Personality Item Pool (IPIP) "Big-Five" Factor Markers). [20]

Το ερωτηματολόγιο αποτελείται από 44 ερωτήσεις και ο ερωτούμενος πρέπει να επιλέξει μια από τις 5 επιλογές.

Κλίμακα	Διαφωνώ κάθετα	Διαφωνώ ελάχιστα	Δεν Διαφωνώ/ Δεν συμφωνώ	Συμφωνώ ελάχιστα	Συμφωνώ κάθετα
Βαθμολογία	1	2	3	4	5

Πίνακας 1.2 Κλίμακα - Big Five Inventory

Οι ερωτήσεις έχουν προκαθορισμένη σειρά και αναφέρονται στην Εικόνα 1.3.

I see Myself as Someone Who...

- | | |
|--|---|
| ___ 1. Is talkative | ___ 23. Tends to be lazy |
| ___ 2. Tends to find fault with others | ___ 24. Is emotionally stable, not easily upset |
| ___ 3. Does a thorough job | ___ 25. Is inventive |
| ___ 4. Is depressed, blue | ___ 26. Has an assertive personality |
| ___ 5. Is original, comes up with new ideas | ___ 27. Can be cold and aloof |
| ___ 6. Is reserved | ___ 28. Perseveres until the task is finished |
| ___ 7. Is helpful and unselfish with others | ___ 29. Can be moody |
| ___ 8. Can be somewhat careless | ___ 30. Values artistic, aesthetic experiences |
| ___ 9. Is relaxed, handles stress well | ___ 31. Is sometimes shy, inhibited |
| ___ 10. Is curious about many different things | ___ 32. Is considerate and kind to almost everyone |
| ___ 11. Is full of energy | ___ 33. Does things efficiently |
| ___ 12. Starts quarrels with others | ___ 34. Remains calm in tense situations |
| ___ 13. Is a reliable worker | ___ 35. Prefers work that is routine |
| ___ 14. Can be tense | ___ 36. Is outgoing, sociable |
| ___ 15. Is ingenious, a deep thinker | ___ 37. Is sometimes rude to others |
| ___ 16. Generates a lot of enthusiasm | ___ 38. Makes plans and follows through with them |
| ___ 17. Has a forgiving nature | ___ 39. Gets nervous easily |
| ___ 18. Tends to be disorganized | ___ 40. Likes to reflect, play with ideas |
| ___ 19. Worries a lot | ___ 41. Has few artistic interests |
| ___ 20. Has an active imagination | ___ 42. Likes to cooperate with others |
| ___ 21. Tends to be quiet | ___ 43. Is easily distracted |
| ___ 22. Is generally trusting | ___ 44. Is sophisticated in art, music, or literature |

Εικόνα 1.2 Ερωτήσεις Big Five Inventory

Για τον υπολογισμό της βαθμολογίας για την κάθε διάσταση, λαμβάνονται υπόψη συγκεκριμένες ερωτήσεις. Ο υπολογισμός βαθμολογίας της κάθε διάστασης προκύπτει από τον

πίνακα της εικόνας 1.4. Όταν δίπλα στον αριθμό της ερώτησης εμφανίζεται το σύμβολο «R», αυτό σημαίνει την αντιστροφή της βαθμολογίας για τη συγκεκριμένη ερώτηση.

Extraversion: 1, 6R, 11, 16, 21R, 26, 31R, 36
Agreeableness: 2R, 7, 12R, 17, 22, 27R, 32, 37R, 42
Conscientiousness: 3, 8R, 13, 18R, 23R, 28, 33, 38, 43R
Neuroticism: 4, 9R, 14, 19, 24R, 29, 34R, 39
Openness: 5, 10, 15, 20, 25, 30, 35R, 40, 41R, 44

Εικόνα 1.3 Υπολογισμός βαθμολογίας (Big Five Factors)

1.4.3 Διαφορετικά Ψυχολογικά Τεστ

- **Sport-Confidence Inventory (SCI):** Είναι ένα ερωτηματολόγιο αυτοαξιολόγησης που χρησιμοποιείται για την αξιολόγηση της εμπιστοσύνης ενός ατόμου στις αθλητικές του ικανότητες. Το SCI μετρά διάφορες πτυχές της εμπιστοσύνης, συμπεριλαμβανομένης της γενικής αυτοπεποίθησης, της εμπιστοσύνης του αθλητή στις φυσικές του ικανότητες και της εμπιστοσύνης που έχει στις δεξιότητες του που απαιτούνται κατά τη διάρκεια του αγώνα.
- **Athletic Identity Measurement Scale (AIMS):** Είναι ένα ερωτηματολόγιο αυτοαξιολόγησης που χρησιμοποιείται για την αξιολόγηση της αθλητικής ταυτότητας ενός ατόμου. Η αθλητική ταυτότητα αναφέρεται στον βαθμό στον οποίο ένα άτομο ταυτίζεται ως αθλητής και θεωρείται σημαντική πτυχή της αυτοαντίληψης ενός ατόμου. Έρευνες έχουν δείξει ότι η αθλητική ταυτότητα συνδέεται άμεσα με τα εσωτερικά κίνητρα, την επιμονή και την ευημερία των αθλητών, ενώ η αρνητική αθλητική ταυτότητα συνδέεται με τη σωματική εξουθένωση, το άγχος και άλλα προβλήματα ψυχικής υγείας.
- **Competitive State Anxiety Inventory-2 (CSAI-2):** Είναι ένα ερωτηματολόγιο αυτοαξιολόγησης που χρησιμοποιείται για την αξιολόγηση του επιπέδου άγχους ενός ατόμου σε ένα ανταγωνιστικό αθλητικό περιβάλλον. Το αγωνιστικό άγχος αναφέρεται στο επίπεδο ανησυχίας και άγχους που βιώνει ένας αθλητής πριν και κατά τη διάρκεια του αγώνα. Μπορεί να χρησιμοποιηθεί ως μέσο παρακολούθησης των αλλαγών στα επίπεδα άγχους με την πάροδο του χρόνου και αξιολόγησης της αποτελεσματικότητας διαφόρων στρατηγικών για τη μείωση του άγχους στους αθλητές.
- **Task and Ego Orientation in Sport Questionnaire (TEOSQ):** είναι ένα ερωτηματολόγιο αυτοαξιολόγησης που χρησιμοποιείται για την αξιολόγηση των κινήτρων ενός ατόμου στον αθλητισμό. Το ερωτηματολόγιο μετρά δύο διαστάσεις των κινήτρων: τον «προσανατολισμό ως προς στο έργο» (task orientation) και τον «προσανατολισμό στον/εαυτό του» (ego orientation). Ο προσανατολισμός στο έργο αναφέρεται όταν το άτομο εστιάζει στη μάθηση και επιθυμεί να βελτιώσει τη φυσική του κατάσταση, ταυτόχρονα το κύριο κίνητρο του αθλητή είναι η επίτευξη κάποιου συγκεκριμένου στόχου και η ικανοποίηση εσωτερικών αναγκών. Τα άτομα με υψηλό «προσανατολισμό στο έργο» είναι πιο πιθανό να προσεγγίσουν τον αθλητισμό ως πρόκληση και να επιμείνουν μπροστά σε αποτυχίες. Ο προσανατολισμός στο εγώ, από την άλλη πλευρά, αναφέρεται στην εστίαση στην επίδειξη ικανότητας και στην υπεροχή έναντι των άλλων. Τα άτομα με υψηλό «προσανατολισμό στο

Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων εγώ/εαυτό τους» είναι πιο πιθανό να προσεγγίσουν τον αθλητισμό ως μέσο για να εδραιώσουν την υπεροχή τους έναντι των άλλων.

1.5 Συστήματα παρακολούθησης απόδοσης αθλητή

Υπάρχουν πολλά συστήματα παρακολούθησης απόδοσης αθλητή, ορισμένα παραδείγματα περιγράφονται παρακάτω.

- **Garmin Forerunner:** Ένα ρολόι παρακολούθησης που παρέχει λεπτομερείς μετρήσεις για την απόδοση του αθλητή σε πολλές δραστηριότητες, όπως τρέξιμο, ποδηλασία και κολύμβηση. Οι μετρήσεις λαμβάνονται από αισθητήρα στον καρπό (ή από τη ζώνη μέσης που είναι συνδεδεμένη με το ρολόι) και ανάλογα με αυτά παρέχονται συμπεράσματα.

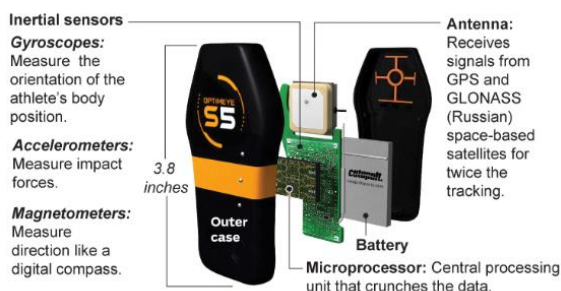


Εικόνα 1.4 Garmin Forerunner 945 I



Εικόνα 1.5 Garmin Forerunner 945 II

- **Catapult:** Μια πλατφόρμα παρακολούθησης που χρησιμοποιεί αισθητήρες για τη συλλογή δεδομένων σχετικά με την απόδοση του αθλητή κατά τη διάρκεια αγώνων και προπονήσεων. Χρησιμοποιείται ιδιαίτερα στο ποδόσφαιρο και σε άλλα ομαδικά αθλήματα.



Εικόνα 1.6 Catapult 1



Εικόνα 1.7 Catapult 2

2 ΚΕΦΑΛΑΙΟ 2^ο : Τεχνικό υπόβαθρο

2.1 Εισαγωγή κεφαλαίου

Σε αυτό το κεφάλαιο αναλύεται το τεχνικό υπόβαθρο που είναι αναγκαίο για την υλοποίηση την εφαρμογής. Στην ενότητα 2.2 παρουσιάζεται η γλώσσα προγραμματισμού Kotlin, εξηγούνται τα πλεονεκτήματα της και αναφέρονται παραδείγματα σύγχρονων εφαρμογών που υλοποιήθηκαν με τη χρήση της. Επιπλέον, στην ενότητα 2.3 περιγράφεται η δομή του Android Studio και αναλύονται τα βασικά μέρη του.

Τέλος, στην ενότητα 2.4 παρουσιάζεται η πλατφόρμα της Firebase η οποία χρησιμοποιήθηκε στην εφαρμογή ως βάση δεδομένων. Αναλύονται τα πλεονεκτήματα της πλατφόρμας και δυνατότητες που προσφέρει και παρέχονται οι απαραίτητες πληροφορίες για τη σύνδεση της με το πρόγραμμα Android Studio.

2.2 Γλώσσα προγραμματισμού Kotlin

2.2.1 Εισαγωγή

Η Kotlin είναι μια cross-platform, statically-typed γλώσσα προγραμματισμού που παρουσιάστηκε για πρώτη φορά από την εταιρία JetBrains το 2011. Έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι πλήρως συμβατή με τη Java, γεγονός που την καθιστά ιδανική επιλογή για την ανάπτυξη εφαρμογών Android [1].

Είναι μια εξαιρετική επιλογή για τη δημιουργία σύγχρονων και δυναμικών εφαρμογών. Η κοινότητα προγραμματιστών που χρησιμοποιεί την Kotlin διαρκώς αυξάνεται. Έχει υιοθετηθεί από πολλούς μεγάλους οργανισμούς, συμπεριλαμβανομένου και της Google, η οποία το 2017 όρισε την Kotlin ως επίσημη γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών Android.

Πλεονεκτήματα της Kotlin

- **Concise Syntax (συνοπτική σύνταξη):**

Η σύνταξη της είναι αρκετά απλή σε σχέση με αυτή που έχει η Java. Επιτρέπει την ίδια λειτουργικότητα με λιγότερες γραμμές κώδικα, με αποτέλεσμα την ταχύτερη ανάπτυξη και την ευκολότερη συντήρηση.

- **Null Safety:**

Η Kotlin διαθέτει ενσωματωμένα χαρακτηριστικά (null safety features) για την αποφυγή που βοηθούν στην αποφυγή σφαλμάτων που προκύπτουν από μη ηθελημένο μηδενισμό κάποιων μεταβλητών, που αποτελεί μια κοινή πηγή σφαλμάτων στη Java. Με αυτόν τον τρόπο, οι χρήστες δεν χρειάζονται να ελέγχουν την κενή είσοδο και με αυτό τον τρόπο βελτιώνεται η αξιοπιστία του κώδικα.

- **Interoperability (διαλειτουργικότητα):**

Η Kotlin είναι πλήρως συμβατή με τη Java, πράγμα που σημαίνει ότι επιτρέπεται η χρήση βιβλιοθηκών της Java στον κώδικα Kotlin (και το αντίστροφο). Αυτό καθιστά εύκολη τη σταδιακή υιοθέτηση της Kotlin και την ενσωμάτωσή της σε ήδη υπάρχουσες εφαρμογές που υλοποιήθηκαν με τη χρήση της Java.

- **Functional Programming:** Η Kotlin υποστηρίζει δομές λειτουργικού προγραμματισμού, όπως ανώνυμες διαδικασίες που μπορούν να χρησιμοποιηθούν ως κοινές τιμές (lambdas), συναρτήσεις ανώτερης τάξης (higher-order functions) και τη δυνατότητα της διατήρησης στοιχείων ως αμετάβλητων (immutability), οι οποίες κάνουν τον κώδικα πιο συνοπτικό και εκφραστικό.
- **Coroutines:**
Η Kotlin έχει ενσωματωμένη υποστήριξη για coroutines, η δυνατότητα αυτή αποτελεί ένα ισχυρό εργαλείο για τη συγγραφή ασύγχρονου, κώδικα (non-blocking) που αποτρέπει την εμπλοκή άλλων λειτουργιών. Τα coroutines απλοποιούν τον χειρισμό των ασύγχρονων εργασιών και διευκολύνουν τη συγγραφή κώδικα που είναι τόσο αποδοτικός όσο και ευανάγνωστος.
- **Data Classes:**
Παρέχει έναν συνοπτικό τρόπο για τον ορισμό κλάσεων δεδομένων, ο οποίος βοηθάει στη μείωση του κώδικα κατά την εργασία με αντικείμενα δεδομένων. Οι κλάσεις δεδομένων μπορούν να παράγουν αυτόματα μεθόδους όπως equals(), hashCode() και toString() με βάση τις ιδιότητές τους.

2.2.2 Δημοφιλής εφαρμογές που χρησιμοποιούν την Kotlin

Η γλώσσα προγραμματισμού Kotlin, λόγω των πολλών πλεονεκτημάτων της, χρησιμοποιείται για την υλοποίηση εφαρμογών σε αρκετούς και διαφορετικούς τομείς. Παρακάτω περιγράφονται κάποιες από τις πιο γνωστές εφαρμογές, που τα τελευταία χρόνια άρχισαν να εφαρμόζουν την Kotlin και βελτιώνουν τη λειτουργία τους με τη βοήθεια της.

Uber

Η Uber είναι μια πλατφόρμα μετακίνησης που συνδέει τους επιβάτες με κοντινούς οδηγούς μέσω μιας εφαρμογής για κινητά τηλέφωνα. Χρησιμοποιώντας την εφαρμογή Uber, οι επιβάτες μπορούν να ζητήσουν μια διαδρομή προς την επιθυμητή τοποθεσία τους και ένας κοντινός οδηγός θα βρεθεί μαζί τους για να τους προσφέρει τη διαδρομή. Η εφαρμογή παρέχει στους επιβάτες πληροφορίες σχετικά με τον οδηγό, το όχημα και τον εκτιμώμενο χρόνο άφιξης. Το κόμιστρο για τη διαδρομή υπολογίζεται με βάση την απόσταση, τον χρόνο και τη ζήτηση και χρεώνεται απευθείας στη μέθοδο πληρωμής του επιβάτη που είναι καταχωρημένη στο αρχείο της Uber.

Δραστηριοποιείται σε περισσότερες από 600 πόλεις παγκοσμίως και προσφέρει διάφορους τύπους διαδρομών, όπως κανονικά αυτοκίνητα, πολυτελή οχήματα και κοινές διαδρομές. Η Uber, έχει επίσης επενδύσει στην ανάπτυξη εφαρμογών Kotlin για διαδικασίες Internal Tooling, όπως επεξεργαστές σχολιασμού(annotation processors) και Grandle plugins.

Airbnb

Η Airbnb είναι η κορυφαία πλατφόρμα παγκοσμίως στην εύρεση καταλυμάτων. Η πλατφόρμα επιτρέπει στους οικοδεσπότες να καταχωρούν τα ακίνητά τους, να καθορίζουν τις δικές τους τιμές και να επικοινωνούν με τους πιθανούς επισκέπτες. Οι επισκέπτες μπορούν να αναζητήσουν διαθέσιμα καταλύματα με βάση τις ημερομηνίες του ταξιδιού τους, την επιθυμητή

τοποθεσία και να κάνουν κράτηση για τη διαμονή τους απευθείας μέσω της πλατφόρμας Airbnb. Η Airbnb χρησιμοποίησε την Kotlin σε ένα νέο πλαίσιο προϊόντων για Android (MvRx).

Netflix

Το Netflix είναι η κορυφαία υπηρεσία streaming ψυχαγωγίας στον κόσμο με 183 εκατομμύρια συνδρομητές σε 190 χώρες. Έχει ανακατασκευάσει το τρέχον UI player στην εφαρμογή Netflix Android χρησιμοποιώντας εξ' ολοκλήρου την Kotlin.

Pinterest

Το Pinterest είναι μια πλατφόρμα κοινωνικής δικτύωσης που επιτρέπει στους χρήστες να ανακαλύπτουν και να αποθηκεύουν ιδέες για διάφορα θέματα. Επιτρέπει τη δημιουργία εικονικών πινάκων (pinboards) με την επισύναψη εικόνων, βίντεο και περιεχομένου από το διαδίκτυο ή από άλλους χρήστες του Pinterest. Η πλατφόρμα χρησιμοποιεί τεχνολογία οπτικής αναζήτησης. Η εταιρία έχει ορίσει την Kotlin ως κύρια γλώσσα προγραμματισμού για την ανάπτυξη των Android εφαρμογών της.

2.3 Android Studio

2.3.1 Εισαγωγή

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android, βασισμένο στο IntelliJ IDEA. Κυκλοφόρησε για πρώτη φορά το 2013 από την Google και έκτοτε έχει γίνει το πιο ευρέως χρησιμοποιούμενο IDE για την ανάπτυξη εφαρμογών Android. Παρέχει ένα πλήρες περιβάλλον για τη δημιουργία εφαρμογών Android υψηλής ποιότητας και απόδοσης, ενώ ενημερώνεται τακτικά με νέα χαρακτηριστικά (features) και διορθώσεις σφαλμάτων.

2.3.2 Δομή του Android Studio

Manifest file

Το αρχείο manifest απαιτείται σε κάθε εφαρμογή Android και είναι το πρώτο αρχείο που διαβάζει το σύστημα κατά την εγκατάσταση της εφαρμογής. Το σύστημα χρησιμοποιεί τις πληροφορίες στο αρχείο manifest για να καθορίσει την ταυτότητα της εφαρμογής, τα δικαιώματα ασφαλείας και τα στοιχεία που μπορούν να εκκινήσουν από άλλες εφαρμογές. Είναι ένα αρχείο XML που παρέχει βασικές πληροφορίες στο λειτουργικό σύστημα Android σχετικά με την εφαρμογή που εκτελείται. Περιλαμβάνει πληροφορίες όπως το όνομα του πακέτου της εφαρμογής (app's packages name), τα δικαιώματα (permissions, π.χ. Internet permission), πληροφορίες για όλες τις δραστηριότητες (activities), τις υπηρεσίες (services), τους παραλήπτες (receivers) και τα φίλτρα προθέσεων (intent filters).

Το αρχείο manifest βρίσκεται συνήθως στον ριζικό κατάλογο ενός έργου Android και ονομάζεται "AndroidManifest.xml". Μπορεί να επεξεργαστεί χρησιμοποιώντας έναν επεξεργαστή κειμένου ή τον οπτικό επεξεργαστή που είναι ενσωματωμένος στο Android Studio.

Layout file

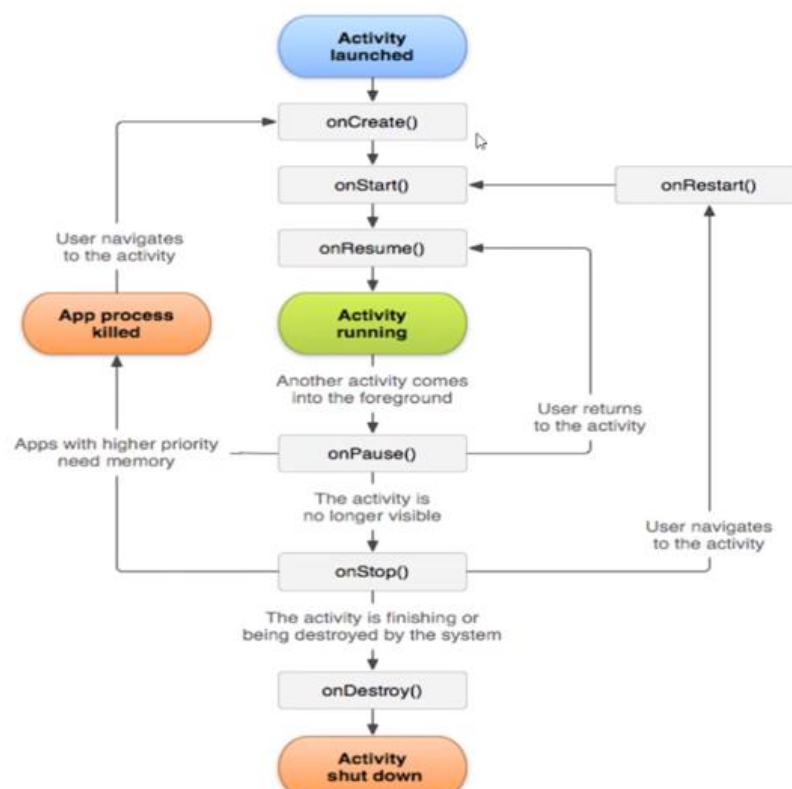
Το layout file είναι ένα αρχείο XML που ορίζει τη διεπαφή χρήστη για μια δραστηριότητα ή ένα τμήμα. Περιγράφει τη διάταξη και την εμφάνιση των UI στοιχείων (user interface components), όπως προβολές κειμένου (text views), κουμπιά (buttons), εικόνες (images) και πεδία εισαγωγής (user input fields – edit texts), που θα εμφανίζονται στην οθόνη. Όταν δημιουργείται μια δραστηριότητα (activity) ή ένα fragment, δημιουργείται αυτόματα και το αντίστοιχο αρχείο διάταξης (layout file) για τη δημιουργία της διεπαφής χρήσης (user interface) και την εμφάνισή της στην οθόνη.

Τα αρχεία layout δημιουργούνται στον φάκελο "res/layout" σε μια εφαρμογή Android και μπορούν να επεξεργαστούν χρησιμοποιώντας έναν επεξεργαστή κειμένου ή τον οπτικό επεξεργαστή του Android Studio. Ο οπτικός επεξεργαστής παρέχει τη δυνατότητα εισαγωγής δεδομένων με την επιλογή drag-and-drop για την προσθήκη και διάταξη UI στοιχείων και παράγει αυτόματα τον αντίστοιχο κώδικα XML. Κάθε αρχείο διάταξης έχει ένα βασικό στοιχείο (root element), το οποίο μπορεί να είναι ένας τύπος διάταξης (Linear Layout, Constraint layout, Relative Layout), ή ένα στοιχείο UI (text, color, size, position). Το βασικό στοιχείο μπορεί να περιέχει άλλα στοιχεία που καθορίζουν τις ιδιότητες και τα χαρακτηριστικά των στοιχείων UI, όπως το κείμενο, το χρώμα, το μέγεθος και τη θέση.

Java file

Όταν δημιουργείτε ένα νέο έργο στο Android Studio, δημιουργεί αυτόματα ένα αρχείο Java που ονομάζεται MainActivity.kt, το οποίο χρησιμεύει ως σημείο εισόδου για την εφαρμογή σας. Το αρχείο MainActivity.kt (και κάθε νέο αρχείο δραστηριότητας) περιέχει την κύρια κλάση για την εφαρμογή Android και είναι υπεύθυνο για το χειρισμό διαφόρων συμβάντων της εφαρμογής, όπως onCreate(), onStart(), onResume(), onPause(), onStop() και onDestroy(). Αυτές οι μέθοδοι καλούνται από το σύστημα Android σε διάφορα στάδια του κύκλου ζωής της εφαρμογής (Εικόνα 2.1).

Στο φάκελο java βρίσκονται τα αρχεία στα οποία ορίζεται η λειτουργία της κάθε δραστηριότητας (activity) της εφαρμογής.



Gradle file

Στο Android Studio, υπάρχουν δύο αρχεία build.gradle: ένα για το έργο (project) και ένα για την ενότητα της εφαρμογής (app module).

▪ **Gradle file (project level):**

Το αρχείο build.gradle σε επίπεδο έργου βρίσκεται στον κατάλογο αρχείων του έργου και χρησιμοποιείται για τη διαμόρφωση ρυθμίσεων που ισχύουν για ολόκληρο το έργο, όπως οι εκδόσεις των πρόσθετων Gradle και των αποθετηρίων buildscript (buildscript repositories).

▪ **Gradle file (app module):**

Αυτό το αρχείο είναι υπεύθυνο για τη διαμόρφωση της διαδικασίας κατασκευής της συγκεκριμένης εφαρμογής. Το αρχείο app Gradle ορίζει τις εξαρτήσεις της εφαρμογής (dependencies), τους τύπους κατασκευής (build types), τα flavors που χρησιμοποιούνται για τη δημιουργία διαφορετικών εκδόσεων των εφαρμογών. Καθορίζει επίσης την ελάχιστη και τις στοχευόμενες εκδόσεις SDK, την έκδοση της γλώσσας προγραμματισμού (version code) και άλλα χαρακτηριστικά που χρησιμοποιούνται για τη δημιουργία του αρχείου APK.

Το αρχείο Gradle της εφαρμογής περιλαμβάνει συνήθως τα ακόλουθα τμήματα:

- **Plugins:** Ορίζει τα πρόσθετα που χρησιμοποιούνται για την κατασκευή της εφαρμογής, όπως το πρόσθετο Android Gradle (π.χ. 'com.android.application').
- **Android:** Περιέχει τις ρυθμίσεις κατασκευής της εφαρμογής, όπως οι ελάχιστες και οι «στοχευμένες» εκδόσεις SDK (minimum and target SDK versions), οι τύποι κατασκευής (build types) και οι «παραλλαγές» προϊόντων (product flavors).
- **Dependencies:** Αυτή η ενότητα παραθέτει τις εξωτερικές βιβλιοθήκες από τις οποίες εξαρτάται η εφαρμογή, όπως η βιβλιοθήκη υποστήριξης Android (implementation 'com.google.firebase:firebase-auth-ktx:21.1.0').
- **BuildTypes:** Αυτή η ενότητα ορίζει τους διαφορετικούς τύπους κατασκευής για την εφαρμογή, όπως debug και release.
- **ProductFlavors:** Αυτή η ενότητα ορίζει τις διαφορετικές επιλογές του προϊόντος για την εφαρμογή, όπως η δωρεάν και η επί πληρωμή έκδοση.

Theme file

Το αρχείο θέματος σε μια εφαρμογή Android είναι ένα αρχείο XML που ορίζει το οπτικό στυλ και την εμφάνιση των στοιχείων του περιβάλλοντος εργασίας της εφαρμογής. Περιέχει πληροφορίες για τα χρώματα, τις γραμματοσειρές και άλλα χαρακτηριστικά που συνδέονται με μια συνεπής εμφάνιση και αίσθηση σε όλη την εφαρμογή.

Ένα αρχείο θέματος χρησιμοποιείται για τον ορισμό του προεπιλεγμένου στυλ της εφαρμογής, αλλά μπορεί επίσης να προσαρμοστεί για συγκεκριμένες δραστηριότητες ή στοιχεία UI. Το αρχείο θέματος αποθηκεύεται συνήθως στον φάκελο "res/values" του έργου και ονομάζεται "styles.xml". Το σύστημα Android παρέχει ένα σύνολο προκαθορισμένων θεμάτων, όπως τα

"Theme.AppCompat" και "Theme.Material", τα οποία μπορούν να χρησιμοποιηθούν ως σημείο αναφοράς για την προσαρμογή της εμφάνισης της εφαρμογής. Υπάρχει η δυνατότητα δημιουργίας νέων διαφορετικών θεμάτων ανάλογα με τις απαιτήσεις της εφαρμογής.

String file

Ένα αρχείο συμβολοσειρών (string file), επίσης γνωστό ως αρχείο πόρων συμβολοσειρών (string resource file), σε μια εφαρμογή Android περιέχει όλες τις συμβολοσειρές που χρησιμοποιούνται στην εφαρμογή. Χρησιμοποιώντας resources files επιτρέπεται ο διαχωρισμός το περιεχομένου της εφαρμογής από την παρουσίασή της, γεγονός που διευκολύνει τη συντήρηση και την ενημέρωση του περιεχομένου που εμφανίζονται στην εφαρμογή. Χρησιμοποιείται ιδιαίτερα όταν η εφαρμογή υποστηρίζει συγχρόνως παραπάνω από μια γλώσσα.

Στο Android, οι string resources ορίζονται σε αρχεία XML που βρίσκονται στον κατάλογο res/values της εφαρμογής. Από προεπιλογή, υπάρχει ένα αρχείο που ονομάζεται strings.xml το οποίο δημιουργείται αυτόματα με τη δημιουργία ενός νέου έργου Android.

2.3.3 Εφαρμογές Android για την παρακολούθηση αθλητικής δραστηριότητας

Οι εφαρμογές Android για την παρακολούθηση αθλητικής δραστηριότητας είναι ιδιαίτερα διαδεδομένες εφαρμογές που βοηθούν τους χρήστες να καταγράψουν και να αναλύσουν την αθλητική τους δραστηριότητα, όπως το τρέξιμο, η ποδηλασία, και άλλες φυσικές δραστηριότητες. Αυτές οι εφαρμογές συνήθως παρέχουν διάφορα χαρακτηριστικά, όπως καταγραφή της απόστασης που διανύθηκε, του χρόνου, της ταχύτητας, υψομετρικών διαφορών και άλλων μετρήσεων.

Μερικά παραδείγματα εφαρμογών Android για την παρακολούθηση αθλητικής δραστηριότητας περιγράφονται παρακάτω:

Strava: Η εφαρμογή Strava είναι μια από τις πιο δημοφιλείς εφαρμογές παρακολούθησης αθλητικής δραστηριότητας. Παρέχει δυνατότητες καταγραφής μεταξύ όλων των αποστάσεων που διανύθηκαν, του χρόνου, του μέσου και μέγιστου ρυθμού, των θερμίδων που καταναλώθηκαν κατά τη διάρκεια της προσπάθειας. Επιπλέον, επιτρέπεται η κοινωνική συναναστροφή, όπως παρακολούθηση δραστηριότητας άλλων χρηστών, σχολιασμός, κλπ.

Nike Run Club: Είναι μια ολοκληρωμένη εφαρμογή τρεξίματος που παρέχει στους χρήστες όλα τα δεδομένα που χρειάζονται για να παρακολουθούν, να βελτιώνουν και να απολαμβάνουν την εμπειρία του τρεξίματος. Προσφέρει μια ποικιλία χαρακτηριστικών, όπως παρακολούθηση GPS, εξατομικευμένα προγράμματα προπόνησης, προσαρμοσμένες λίστες μουσικής αναπαραγωγής και κοινωνική συναναστροφή. Οι χρήστες μπορούν επίσης να συμμετάσχουν σε προκλήσεις και δοκιμασίες και να συναγωνιστούν με άλλους δρομείς απ' όλο τον κόσμο.

2.4 Βάση δεδομένων Firebase

Σε αυτή την υποενότητα περιγράφεται η πλατφόρμα Firebase, η χρησιμότητα της, οι δυνατότητες που δίνει στους προγραμματιστές αλλά και στους χρήστες εφαρμογών που υλοποιήθηκαν με τη βοήθεια της.

2.4.1 Εισαγωγή – Τι είναι η Firebase

Είναι μια ολοκληρωμένη πλατφόρμα που διευκολύνει τους προγραμματιστές να δημιουργούν και να διαχειρίζονται τις εφαρμογές τους γρήγορα και αποτελεσματικά. Η Firebase είναι μια

Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων πλατφόρμα Backend-as-a-Service (BaaS) που παρέχει ένα ευρύ φάσμα υπηρεσιών βασισμένων στο cloud για τη δημιουργία και την ανάπτυξη εφαρμογών ιστού (web applications) αλλά και εφαρμογών κινητών τηλεφώνων (mobile applications). Αναπτύχθηκε και διαχειρίζεται από την Google [33].

Η Firebase προσφέρει διάφορες επιλογές όπως:

- **Πιστοποίηση ταυτότητας (User authentication):**
Η Firebase παρέχει εύχρηστες υπηρεσίες ελέγχου ταυτότητας για τη διαχείριση εγγραφής νέων χρηστών αλλά και του ελέγχου πρόσβασης των χρηστών.
- **Φιλοξενία (Hosting):**
Το Firebase Hosting παρέχει γρήγορη και ασφαλή φιλοξενία για την εφαρμογή.
- **Λειτουργίες:** Το Firebase Functions είναι ένα περιβάλλον χωρίς διακομιστή για την εκτέλεση κώδικα backend.
- **Αποθηκευτικός χώρος (Storage):** Η Firebase παρέχει αποθηκευτικό χώρο στο cloud για την εφαρμογή, επιτρέποντάς την αποθήκευση περιεχομένου που δημιουργείται από τους χρήστες, όπως εικόνες και βίντεο.
- **Κιτ ML:** Το ML Kit της Firebase είναι ένα πλαίσιο μηχανικής μάθησης που παρέχει προ-εκπαιδευμένα μοντέλα και API για κοινές περιπτώσεις χρήσης, όπως αναγνώριση κειμένου, ανίχνευση προσώπου και άλλα.

Η Firebase προσφέρει δύο επιλογές αποθηκευτικού χώρου: Realtime Database και Cloud Firestore.

- **Βάση δεδομένων σε πραγματικό χρόνο (Realtime Database):**

Η βάση δεδομένων πραγματικού χρόνου είναι μια βάση δεδομένων NoSQL που αποθηκεύει δεδομένα ως έγγραφα JSON. Τα δεδομένα συγχρονίζονται σε πραγματικό χρόνο με όλους τους συνδεδεμένους χρήστες και υποστηρίζει πρόσβαση στα δεδομένα εκτός σύνδεσης. Η Βάση δεδομένων πραγματικού χρόνου είναι κατάλληλη για απλές και γρήγορα αναπτυσσόμενες web και mobile εφαρμογές.

- **To Cloud Firestore:**

Το Cloud Firestore είναι μια βάση δεδομένων NoSQL που βασίζεται σε έγγραφα και παρέχει πιο ισχυρές επιλογές αναζήτησης και δομής δεδομένων σε σύγκριση με τη βάση δεδομένων πραγματικού χρόνου. Υποστηρίζει πρόσβαση χωρίς σύνδεση, έχει καλύτερη κλιμάκωση και μπορεί να διαχειριστεί πιο σύνθετες σχέσεις δεδομένων. Το Cloud Firestore είναι μια καλή επιλογή για μεγαλύτερες, πιο σύνθετες εφαρμογές που απαιτούν προηγμένες δυνατότητες αναζήτησης.

2.4.2 Σύνδεση με το Android Studio

Για να μπορεί να γίνει η σύνδεση του Firebase με μια εφαρμογή που κατασκευάστηκε στο Android Studio είναι απαραίτητο να τηρούνται οι παρακάτω προϋποθέσεις.

- Targets API level 19 (KitKat) or higher
- Uses Android 4.4 or higher
- Uses **Jetpack (AndroidX)**, which includes meeting these version requirements:
 - `com.android.tools.build:gradle` v3.2.1 or later
 - `compileSdkVersion` 28 or later

Εικόνα 2.2 Ελάχιστες προδιαγραφές εφαρμογής

Στη συνέχεια ο χρήστης πρέπει να κατεβάσει το αρχείο «google-services.json» και να το αποθηκεύσει στο **module (app-level)** root directory.

Είναι απαραίτητη η πρόσθεση του παρακάτω κώδικα στους αντίστοιχους υποφακέλους της κάθε εφαρμογής.

Στον φάκελο <project>/build.gradle:

```
dependencies {  
    ...  
  
    // Add the dependency for the Google services Gradle plugin  
    classpath 'com.google.gms:google-services:4.3.15'  
}
```

Εικόνα 2.3 Σύνδεση Firebase – Απαραίτητος κώδικας I

Στον φάκελο <project>/<app-module>/build.gradle :

```
plugins {  
    id 'com.android.application'  
  
    // Add the Google services Gradle plugin  
    id 'com.google.gms.google-services'  
    ...  
}
```

Εικόνα 2.4 Σύνδεση Firebase – Απαραίτητος κώδικας II

Στον φάκελο <project>/<app-module>/build.gradle :

```
dependencies {  
    // ...  
  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:31.2.0')  
  
    // When using the BoM, you don't specify versions in Firebase library dependencies  
  
    // Add the dependency for the Firebase SDK for Google Analytics  
    implementation 'com.google.firebase:firebase-analytics-ktx'  
  
    // TODO: Add the dependencies for any other Firebase products you want to use  
    // See https://firebase.google.com/docs/android/setup#available-libraries  
    // For example, add the dependencies for Firebase Authentication and Cloud Firestore  
    implementation 'com.google.firebase:firebase-auth-ktx'  
    implementation 'com.google.firebase:firebase-firestore-ktx'  
}
```

Εικόνα 2.5 Σύνδεση Firebase – Απαραίτητος κώδικας III

2.4.3 Επαλήθευση χρήστη (User Authentication)

Οι περισσότερες εφαρμογές πρέπει να γνωρίζουν την ταυτότητα ενός χρήστη. Η γνώση της ταυτότητας ενός χρήστη επιτρέπει σε μια εφαρμογή να αποθηκεύει με ασφάλεια τα δεδομένα του χρήστη στο cloud και να παρέχει την ίδια εξατομικευμένη εμπειρία σε όλες τις συσκευές του χρήστη [37].

Το Firebase Authentication παρέχει υπηρεσίες backend, εύχρηστα SDK και έτοιμες βιβλιοθήκες UI για την πιστοποίηση ταυτότητας των χρηστών. Υποστηρίζει τον έλεγχο ταυτότητας χρησιμοποιώντας κωδικούς πρόσβασης, αριθμούς τηλεφώνου, δημοφιλείς μέσα πιστοποίησης ταυτότητας όπως λογαριασμοί στις πλατφόρμες της Google, το Facebook και το Twitter και πολλά άλλα. Το Firebase Authentication ενσωματώνεται στενά με άλλες υπηρεσίες της Firebase και αξιοποιεί βιομηχανικά πρότυπα όπως το OAuth 2.0 και το OpenID Connect, ώστε να μπορεί να ενσωματωθεί εύκολα με το προσαρμοσμένο backend σας [37].

Για να γίνει εγγραφή ενός χρήστη στην εφαρμογή, λαμβάνονται πρώτα τα διαπιστευτήρια ελέγχου ταυτότητας (authentication credentials) από τον χρήστη. Αυτά τα διαπιστευτήρια μπορεί να είναι η διεύθυνση ηλεκτρονικού ταχυδρομείου και ο κωδικός πρόσβασης του χρήστη ή ένα κουπόνι OAuth (OAuth token) από έναν ομοσπονδιακό πάροχο ταυτότητας (για παράδειγμα κατά τη σύνδεση με τη χρήση λογαριασμό Google). Στη συνέχεια, αυτά τα διαπιστευτήρια περνάνε στο Firebase Authentication SDK. Στη συνέχεια, οι backend υπηρεσίες επαληθεύουν αυτά τα διαπιστευτήρια και θα επιστρέφουν μια απάντηση στον χρήστη (αποδοχή ή απόρριψη εισόδου). Μετά την επιτυχή σύνδεση, παρέχεται η πρόσβαση στις βασικές πληροφορίες προφίλ του χρήστη και ελέγχετε η δυνατότητα πρόσβασης του σε δεδομένα που είναι αποθηκευμένα σε άλλα προϊόντα Firebase.

2.4.4 Ανάκτηση και μεταφορά δεδομένων

Για την ανάκτηση δεδομένων από την Firebase Realtime Database, είναι απαραίτητη η δημιουργία μιας Database Reference, η οποία προσδιορίζει το φάκελο (ή υποφάκελο) από τον οποίο «διαβάζεται» η πληροφορία [33].

Στη συνέχεια χρησιμοποιείται η μέθοδος **addValueEventListener()** στην οποία αναφέρεται το DatabaseReference για την ανάκτηση δεδομένων (Εικόνα 2.6).

```

Kotlin+KTX      Java
Android        Android

val postListener = object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        // Get Post object and use the values to update the UI
        val post = dataSnapshot.getValue<Post>()
        // ...
    }

    override fun onCancelled(databaseError: DatabaseError) {
        // Getting Post failed, log a message
        Log.w(TAG, "loadPost:onCancelled", databaseError.toException())
    }
}

postReference.addValueEventListener(postListener)
    
```

Εικόνα 2.6 Ανάκτηση δεδομένων από τη Realtime Database

Για την εγγραφή δεδομένων στη βάση δεδομένων είναι απαραίτητος ο προσδιορισμός του πίνακα ή υποπίνακα (reference) και χρησιμοποιείται η εντολή **setValue()** (Εικόνα 2.7) [33].

```

Kotlin+KTX      Java
Android        Android

fun writeNewUser(userId: String, name: String, email: String) {
    val user = User(name, email)

    database.child("users").child(userId).setValue(user)
}

ReadAndWriteSnippets.kt
    
```

Εικόνα 2.7 Εγγραφή δεδομένων στη Realtime Database

2.4.5 Ασφάλεια της βάσης δεδομένων (Security Rules)

Οι κανόνες ασφαλείας της Firebase είναι ένα σύνολο συνθηκών που καθορίζουν ποιος έχει πρόσβαση στη βάση δεδομένων (Realtime database, Cloud Firestore, Cloud Storage). Είναι ένα ισχυρό εργαλείο για τον έλεγχο της πρόσβασης στα δεδομένα της βάσης δεδομένων με σκοπό τη διασφάλιση προστασίας προσωπικών δεδομένων όλων των χρηστών της εφαρμογής [9]. Χρησιμοποιούνται για την προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση αλλά και για να διασφάλιση της έγκυρης μορφής δεδομένων εισόδου [36].

Οι κανόνες ορίζονται χρησιμοποιώντας μια δηλωτική σύνταξη (declarative syntax), η οποία επιτρέπει τον καθορισμό ποια δεδομένα επιτρέπουν την αναγνώριση και εγγραφή τους και από ποιούς χρήστες είναι αναγνώσιμα και εγγράψιμα από διαφορετικούς τύπους χρηστών ΠΑΔΑ, Τμήμα Η&ΗΜ, Διπλωματική Εργασία, Ματζιάρ Ματέους Κισιτόφ

Αξιολογούνται κάθε φορά που τα δεδομένα διαβάζονται ή γράφονται/προσθέτονται στη βάση δεδομένων. Μπορούν να ενημερώνονται σε πραγματικό χρόνο καθώς εξελίσσεται η εφαρμογή. Η Firebase παρέχει μια κονσόλα που επιτρέπει τον έλεγχο και την προσημείωση της συμπεριφοράς τους υπό κανονικές συνθήκες.

Ορισμένες κοινές περιπτώσεις χρήσης των κανόνων ασφαλείας της Firebase περιλαμβάνουν:

Αυθεντικοποίηση χρηστών (user authentication): Επιτρέπουν τον καθορισμό ποιοι χρήστες μπορούν να διαβάζουν και να προσθέσουν δεδομένα, ανάλογα με το αν έχουν συνδεθεί στην εφαρμογή.

Επικύρωση δεδομένων (validating data): Επιτρέπουν τον έλεγχο της μορφής των δεδομένων, πριν από την εγγραφή τους στη βάση δεδομένων, με σκοπό τη διασφάλιση της συμβατότητας τους με την εφαρμογή.

Προστασία ευαίσθητων δεδομένων (protecting sensitive data): Επιτρέπουν την προστασία ευαίσθητων/ προσωπικών δεδομένων, επιτρέποντας την αναγνώριση/εγγραφή τους μόνο σε συγκεκριμένους χρήστες.

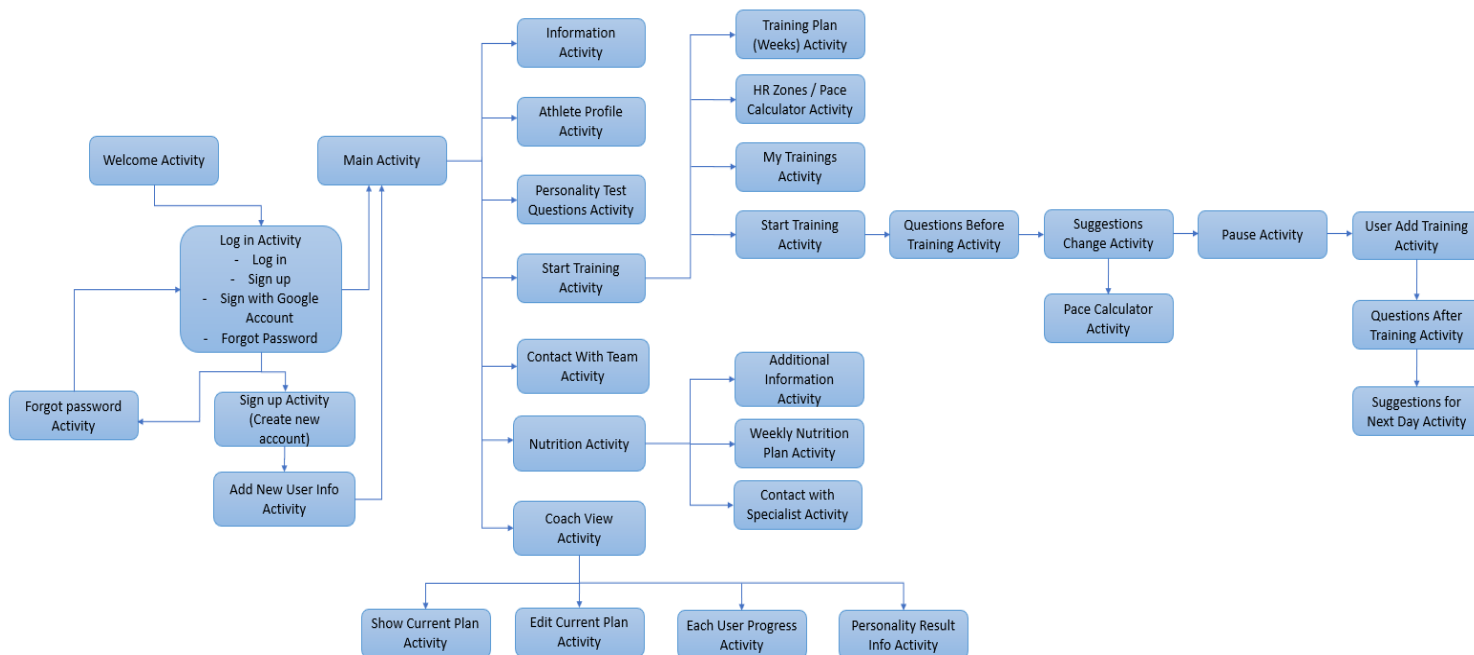
3 ΚΕΦΑΛΑΙΟ 3^ο : Ανάπτυξη εφαρμογής

3.1 Εισαγωγή κεφαλαίου

Σε αυτό το κεφάλαιο περιγράφετε η δομή της εφαρμογής καθοδήγησης δρομέων, καθώς και οι βασικές λειτουργίες της. Επιπλέον, αναλύονται λεπτομερώς τα επιμέρους στοιχεία και εξηγείται ο κώδικας λειτουργίας.

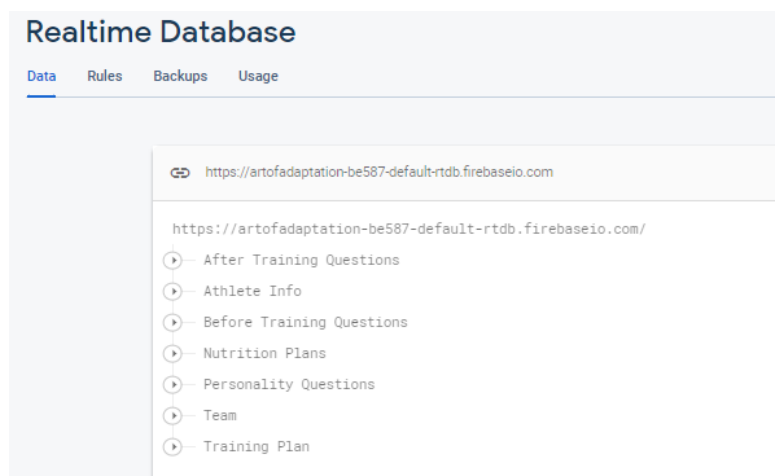
3.2 Δομή της εφαρμογής

Η εφαρμογή αποτελείται από 31 δραστηριότητες (activities) και η σύνδεση τους περιγράφεται στην εικόνα 3.1



Εικόνα 3.1 Δομή της εφαρμογής

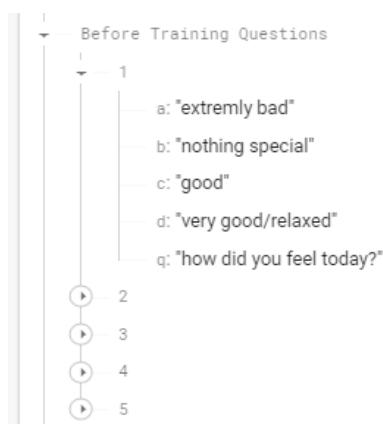
Για την υλοποίηση της βάσης δεδομένων χρησιμοποιήθηκε η πλατφόρμα Firebase της Google και επιλέχθηκε η Realtime Database. Η βασική δομή της παρουσιάζεται στην εικόνα 3.2



Εικόνα 3.2 Δομή βάσης δεδομένων

Ο καθένας από αυτούς τους φακέλους αποτελείται από αρκετούς υποφακέλους (childs), στους οποίους περιέχονται δεδομένα που αφορούν τον χρήστη αλλά και πληροφορίες απαραίτητες για τη λειτουργία της εφαρμογής.

- Στον φάκελο «Before Training Questions» περιέχονται όλες οι ερωτήσεις και οι πιθανές απαντήσεις που καλείται να απαντήσει ο χρήστης πριν από κάθε προπόνηση(Εικόνα 3.3). Στη συνέχεια προκύπτει ένας δείκτης κόπωσης και ανάλογα με τη τιμή του, προτείνονται αλλαγές στην προπόνηση.
- Ο φάκελος «After Training Questions» περιέχει τις ερωτήσεις και τις πιθανές απαντήσεις που καλείται να απαντήσει ο αθλητής μετά την ολοκλήρωση κάθε προπόνησης (Εικόνα 3.4).

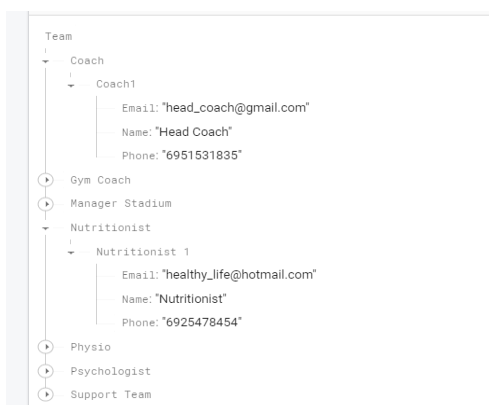


Εικόνα 3.3 Ερωτήσεις πριν την προπόνηση



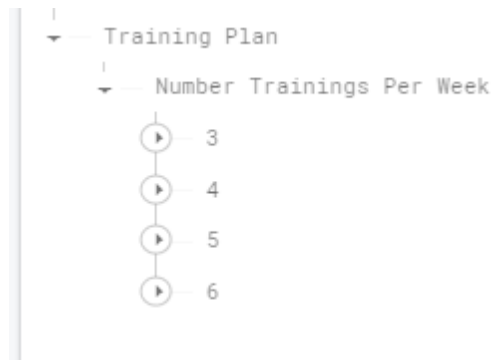
Εικόνα 3.4 Ερωτήσεις μετά την προπόνηση

- Στον φάκελο «Team» περιέχονται τα δεδομένα της ομάδας υποστήριξης του αθλητή (στοιχεία επικοινωνίας όπως ο αριθμός τηλεφώνου και η διεύθυνση ηλεκτρονικού ταχυδρομείου) (Εικόνα 3.5).



Εικόνα 3.5 Ομάδα υποστήριξης

- Ο φάκελος «Training Plan» περιέχει προπονητικά προγράμματα 12 εβδομάδων για κάθε επίπεδο αθλούμενου.(Εικόνα 3.6, Εικόνα 3.7).

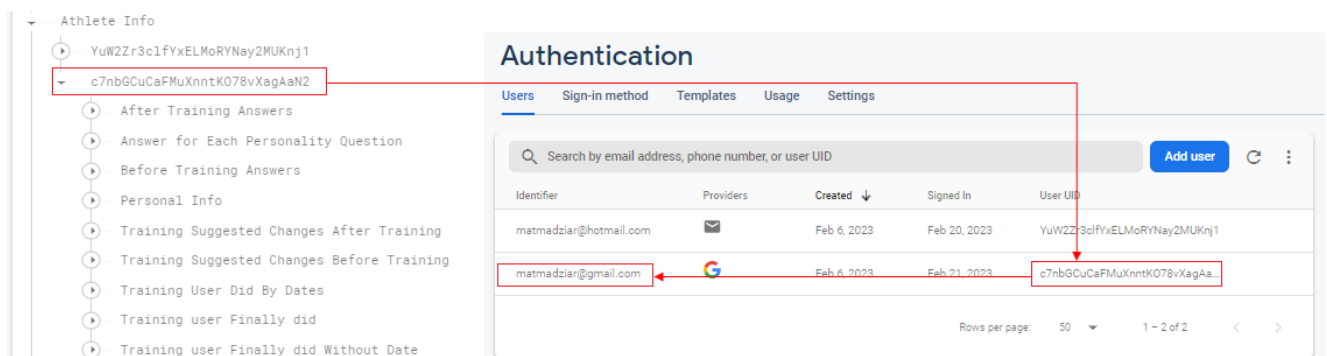


Εικόνα 3.6 Πρόγραμμα προπόνησης I

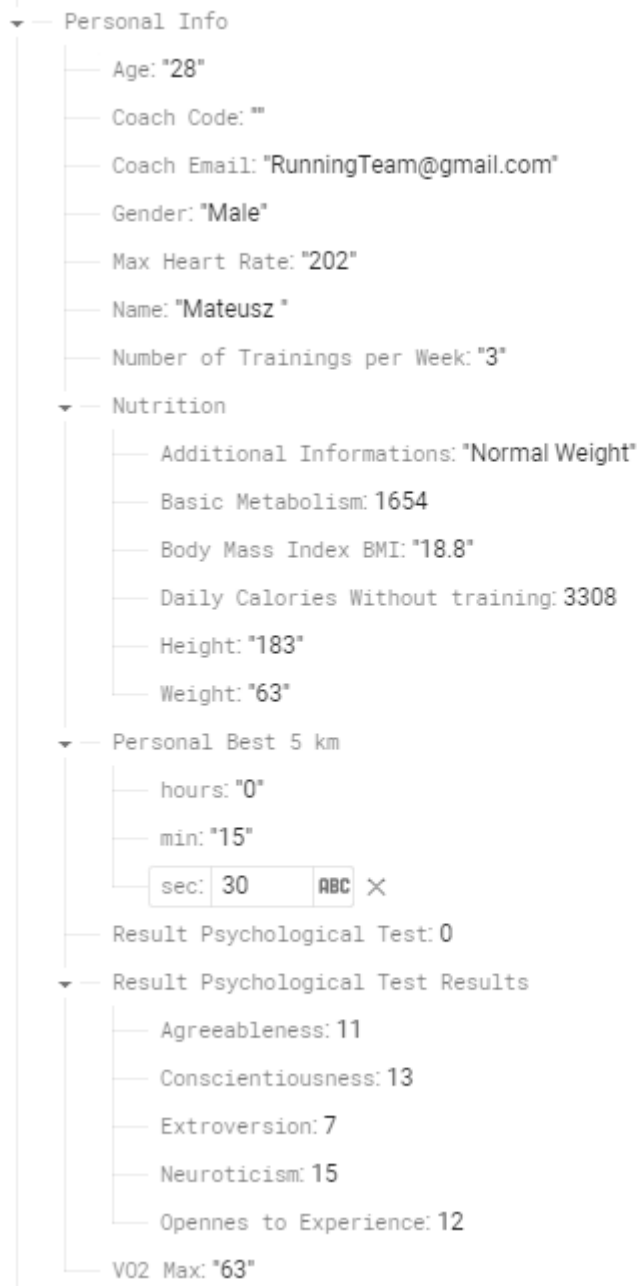


Εικόνα 3.7 Πρόγραμμα προπόνησης II

- Στον φάκελο «Personality Questions» βρίσκονται οι 44 ερωτήσεις που αφορούν τον προσδιορισμό του ψυχολογικού προφίλ του αθλητή με τη μέθοδο Big Five Inventory που αναφέρονται στην ενότητα 1.4.1.
- Ο φάκελος “Athlete Info” περιέχει όλες της πληροφορίες για τον συγκεκριμένο αθλητή. Για κάθε χρήστη δημιουργείται ένας υποφάκελος με όνομα τον μοναδικό κωδικό χρήστη (Εικόνα 3.8) και εκεί καταγράφονται όλες οι πληροφορίες όπως τα βασικά του στοιχεία (όνομα, ηλικία, καλύτερη επίδοση), αλλά και τα σωματομετρικά του χαρακτηριστικά (ύψος, βάρος, ημερήσιες θερμιδικές απαιτήσεις ,κλπ.). Σε αυτόν τον φάκελο αποθηκεύονται και όλες οι απαντήσεις του αθλητή στις ερωτήσεις πριν και μετά την προπόνηση, τα αποτελέσματα του ψυχολογικού τεστ αλλά και οι συμβουλές που έγιναν στον αθλητή με βάση τα ερωτηματολόγια. Επίσης, είναι ο φάκελος στον οποίο αποθηκεύονται όλες οι προπονήσεις που έκανε τελικά ο αθλητής. (Εικόνα 3.9)



Εικόνα 3.8 Εισαγωγή δεδομένων χρήστη



Εικόνα 3.9 Βασικά στοιχεία αθλητή

- Στον φάκελο «Nutrition Plans» περιλαμβάνονται όλα τα ενδεικτικά προγράμματα διατροφής. Κάθε συνδεδεμένος χρήστης έχει πρόσβαση σε αυτά.

3.3 Βασικές λειτουργίες της εφαρμογής

Η εφαρμογή λειτουργεί για min SDK 23 με targed SDK 32, πράγμα το οποίο επιτρέπει τη λειτουργία της στο 98,9% των κινητών με λογισμικό Android.

Οι βασική λειτουργία της εφαρμογής είναι καθοδήγηση δρομέα με την παροχή αυτοματοποιημένης καθοδήγησης. Επιπλέον, παρέχει όλες τις βασικές πληροφορίες στον προπονητή για την κατάσταση όλων των αθλητών του. Οι πληροφορίες που παρέχει η συγκεκριμένη εφαρμογή περιγράφονται παρακάτω.

Προπόνηση:

- Το ενσωματωμένο ερωτηματολόγιο πριν και μετά την προπόνηση επιτρέπει την προσέγγιση του δείκτη κοπώσεως και παρέχονται κάποιες συμβουλές, με σκοπό την ελαχιστοποίηση της πιθανότητας εμφάνισης υπερκόπωσης.
- Παρέχεται προπονητικό πρόγραμμα βασισμένο στις βασικές αρχές προπονητικής που περιγράφονται στο 1^ο κεφάλαιο.
- Παρέχεται η δυνατότητα παρακολούθησης της δραστηριότητας επιλεγμένου αθλητή.
- Επιτρέπεται μόνο στον προπονητή να πραγματοποιεί αλλαγές στο προπονητικό πρόγραμμα και αυτές να φαίνονται σε πραγματικό χρόνο.
- Παρέχεται η δυνατότητα άμεσης επικοινωνίας με την ομάδα υποστήριξης.

Ψυχολογικό προφίλ:

- Παρέχονται πληροφορίες για το ψυχολογικό προφίλ του αθλητή. Τα αποτελέσματα που προκύπτουν παρέχουν σημαντικές πληροφορίες που επιτρέπουν την καλύτερη κατανόηση του τρόπου σκέψης του αθλητή και βοηθάνε στην καλύτερη προσαρμογή της προπόνησης.

Διατροφή:

- Υπολογισμός θερμιδικών αναγκών του αθλούμενου με βάση τα σωματομετρικά χαρακτηριστικά.
- Παρέχεται ενδεικτικό πρόγραμμα διατροφής, προσαρμοσμένο στις θερμιδικές ανάγκες του αθλούμενου.
- Επιτρέπεται η άμεση επικοινωνία με πιστοποιημένο διατροφολόγο.

3.4 Ανάπτυξη εφαρμογής με τη χρήση Android Studio

3.4.1 Διαμόρφωση του Build Grandle file

Για να μπορέσει να επιτευχθεί η σωστή λειτουργία της εφαρμογής είναι απαραίτητη η ένταξη των παρακάτω στοιχείων στον φάκελο “build.gradle(app)”.

▪ View Binding

Επιτρέπει τη σύνδεση των UI στοιχείων συγκεκριμένου layout file απευθείας με το αρχείο Kotlin της κάθε δραστηριότητας. Αυτή η σύνδεση βοηθάει στην αποφυγή συνηθισμένων σφαλμάτων που μπορεί να προκύψουν κατά τη χρήση findViewById(). Αυξάνεται η απόδοση και παράλληλα διευκολύνεται η ανάγνωση του κώδικα.

```
buildFeatures{  
    viewBinding = true  
}
```

Εικόνα 3.10 ViewBinding

▪ Plugins

Το Android Studio, επιτρέπει την πρόσθεση plugins στο φάκελο “BuildGradle.app”, τα οποία μπορούν να χρησιμοποιηθούν για την βελτίωση της λειτουργικότητας του προγράμματος. Αυτά τα plugins είναι διαφορετικά από αυτά που χρησιμοποιούνται για την επέκταση της λειτουργικότητας του IDE.

```

plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'com.google.gms.google-services'
    id 'kotlin-android-extensions'
    id 'kotlin-android'
}
    
```

Εικόνα 3.11 Plugins

- **"com.android.application"**: Αυτό plugin χρησιμοποιείται για τη δημιουργία εφαρμογών Android.
- **"kotlin-android"**: Προσθέτει υποστήριξη για τη γλώσσα προγραμματισμού Kotlin στην εφαρμογή Android.
- **"com.google.gms.google-services"**: Χρησιμοποιείται για την ενσωμάτωση υπηρεσιών της Google, όπως η Firebase, στην εφαρμογή Android.
- **"org.jetbrains.kotlin.android"**: Είναι υπεύθυνο για την ενεργοποίηση της υποστήριξης Kotlin στην εφαρμογή και τη ρύθμιση παραμέτρων του Kotlin Gradle plugin. Παρέχει διάφορα πλεονεκτήματα, όπως υποστήριξη για στη σύνταξη, τη μεταγλώττιση και την αποσφαλμάτωση της γλώσσας προγραμματισμού Kotlin. Βοηθάει επίσης στη μείωση του boilerplate κώδικα και στη βελτίωση της ποιότητας του κώδικα παρέχοντας χαρακτηριστικά όπως null-safety και λειτουργικές προγραμματιστικές δομές.
- **"kotlin-android-extensions"**: Παρέχει ένα σύνολο επεκτάσεων για τη γλώσσα προγραμματισμού Kotlin, που επιτρέπουν την πρόσβασή της στα αρχεία διάταξης (layout files) απευθείας από το αρχείο Kotlin, χωρίς την ανάγκη χρησιμοποίησης της συνάρτησης findViewById().

▪ Dependencies

```

dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.6.0'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-database-ktx:20.1.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    //Declare the dependency for the Firebase Authentication library
    implementation 'com.google.firebase:firebase-auth-ktx:21.1.0'

    //Also declare the dependency for the Google Play services library and specify it's version.
    // For ready google made button (activity login)
    implementation 'com.google.android.gms:play-services-auth:20.1.0'

    implementation platform('com.google.firebase:firebase-bom:31.1.0')
    implementation 'com.google.firebase:firebase-analytics-ktx'

    // For Navigation bar
    implementation 'com.google.android.material:material:1.7.0-alpha04'

    //For cycle image view
    implementation 'de.hdodenhof:circleimageview:3.1.0'
}
    
```

Εικόνα 3.12 Dependencies

3.4.2 Δραστηριότητα Εισόδου

3.4.2.1 Περιγραφή λειτουργίας

Είναι η πρώτη δραστηριότητα της εφαρμογής που ενεργοποιείται όταν ο χρήστης κάνει είσοδο στην εφαρμογή (Εικόνα 3.13).

Εάν ο χρήστης δεν έχει συνδεθεί ξανά από τη συγκεκριμένη συσκευή, μετά το πέρας ορισμένου χρόνου από την εκκίνηση της εφαρμογής, (4 δευτερόλεπτα), μεταφέρεται στην κεντρική δραστηριότητα εισόδου χρήστη (Login Activity) με τη χρήση της συνάρτησης «run» (Εικόνα 3.14).



Εικόνα 3.13 Πρώτη Δραστηριότητα

```

override fun run() {
    //we wants to open the main activity
    val intent = Intent( packageContext: this@WelcomeActivity, LoginActivity ::class.java)
    startActivity(intent) //open the new activity
    finish() //close the activity Welcome
}
    
```

Εικόνα 3.14 Μεταφορά στην κεντρική δραστηριότητα εισόδου

Ο ορισμός της ως “launcher activity” πραγματοποιείται στο Manifest File του Android Studio (Εικόνα 3.15)

```

<activity
    android:name=".WelcomeActivity"
    android:exported="true"
    android:theme="@style/WelcomeThemeSplash">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

Εικόνα 3.15 Δήλωση της πρώτης δραστηριότητας στο Manifest File

Στην περίπτωση που ο χρήστης έχει ξαναχρησιμοποιήσει την εφαρμογή και δεν έχει αποσυνδεθεί μετά την τελευταία χρήση, δεν απαιτείται ξανά η επαλήθευση του με την είσοδο κωδικού, αλλά μεταφέρεται κατευθείαν στην κεντρική δραστηριότητα της εφαρμογής (Main

Activity). Περαιτέρω πληροφορίες για τον τρόπο υλοποίησης αναφέρονται στην υποενότητα 3.4.4 αυτού του κεφαλαίου.

3.4.2.2 Animation

Είναι ένα αρχείο XML στο οποίο περιγράφεται ο τρόπος με τον οποίο εμφανίζονται τα επιμέρους στοιχεία στην οθόνη. Επιτυγχάνεται η σταδιακή εμφάνιση της πρώτης δραστηριότητας και ορίζεται ο τρόπος αλλά και η διάρκεια του.

```
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:duration="3000">
</alpha>
```

Εικόνα 3.16 Splash Animation

Η σύνδεση του με τα στοιχεία του UI (user interface) πραγματοποιείται στον φάκελο Java στο αρχείο “WelcomeActivity.kt”. Ο απαραίτητος κώδικας παρουσιάζεται παρακάτω στην Εικόνα 3.17.

```
package com.example.finalproject
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.view.animation.AnimationUtils
import com.example.finalproject.databinding.ActivityWelcomeBinding

class WelcomeActivity : AppCompatActivity() {
    lateinit var splashBinding: ActivityWelcomeBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        splashBinding = ActivityWelcomeBinding.inflate(layoutInflater)
        val view = splashBinding.root
        setContentView(view)
        //the splash animation (xml file) is assigned to the Alpha animation
        val alphaAnimation = AnimationUtils.loadAnimation(applicationContext, R.anim.splash_anim)

        //We can apply the animation to all components
        splashBinding.textViewSplash.startAnimation(alphaAnimation)
        splashBinding.viewItemBackground.startAnimation(alphaAnimation)
        splashBinding.textViewSplashCorryRights.startAnimation(alphaAnimation)
        //After applying this animation, the Welcome activity should be automatically closed in
        //And the Login Activity should be open
        //We do it by creating a "Handler". Handler allow us to prepare a schedule for the process to be made
        //When the time will pass the Login Activity will be opened
        val handler = Handler(Looper.getMainLooper()) //the parameter list constructor of the handler class is deprecated.
        // For this reason we need to write the constructor parameter
        // we can write an object of the loop or class as a parameter inside teh parenthesis
        handler.postDelayed(object : Runnable {
            override fun run() {
                //we wants to open the main activity
                val intent = Intent(applicationContext, LoginActivity::class.java)
                startActivity(intent) //open the new activity (Login)
                finish() //close the activity Welcome
            }
        }, delayMillis = 4000) // The delay
```

Εικόνα 3.17 Λειτουργία δραστηριότητας εισόδου

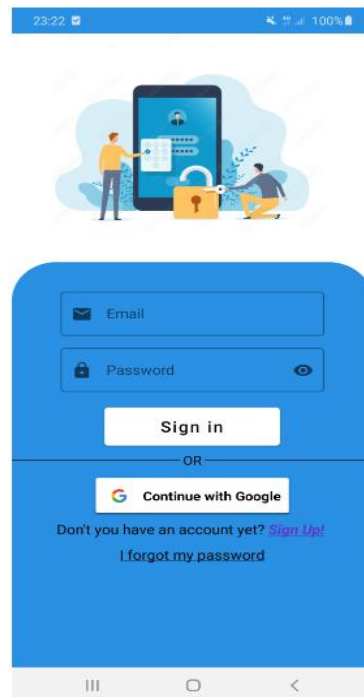
3.4.3 Σύνδεση χρήστη

3.4.3.1 Περίληψη

Αυτή η δραστηριότητα είναι υπεύθυνη για την είσοδο του χρήστη στην εφαρμογή. Οι επιλογές που παρέχονται στον χρήστη είναι οι παρακάτω:

- Είσοδος με τη χρήση ηλεκτρονικού ταχυδρομείου και κωδικού πρόσβασης (επιλογή για χρήστες που διαθέτουν ήδη λογαριασμό).
- Σύνδεση με τη χρήση λογαριασμού Google
- Σύνδεση νέου χρήστη (Δημιουργία λογαριασμού).
- Επιλογή ανάκτησης κωδικού πρόσβασης

Στο πρόγραμμα Android Studio το όνομα της δραστηριότητας είναι "LoginActivity" και το όνομα αρχείου XML είναι "activity_login".



Εικόνα 3.18 Login Activity

Απαραίτητες βιβλιοθήκες:

```

import android.content.Intent
import android.graphics.Color
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.activity.result.ActivityResultCallback
import androidx.activity.result.ActivityResultLauncher
import androidx.activity.result.contract.ActivityResultContracts
import com.example.finalproject.databinding.ActivityLoginBinding
import com.google.android.gms.auth.api.signin.GoogleSignIn
import com.google.android.gms.auth.api.signin.GoogleSignInAccount
import com.google.android.gms.auth.api.signin.GoogleSignInClient
import com.google.android.gms.auth.api.signin.GoogleSignInOptions
import com.google.android.gms.common.api.ApiException
import com.google.android.gms.tasks.Task
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.GoogleAuthProvider
    
```

Εικόνα 3.19 Βιβλιοθήκες LoginActivity

Σύνδεση layout file με το αρχείο Kotlin (View Binding) και οι απαραίτητες δηλώσεις μεταβλητών που χρησιμοποιούνται σε αυτή την δραστηριότητα περιγράφονται στην εικόνα 3.20.

```
class LoginActivity : AppCompatActivity() {
    lateinit var loginBinding: ActivityLoginBinding

    val auth = FirebaseAuth.getInstance() //I need this to use the AuthenticationFirebase - for fun signInUser()

    lateinit var googleSignInClient: GoogleSignInClient // //I needed for Google Sign in button
    lateinit var activityResultLauncher: ActivityResultLauncher<Intent> // for sign in using Google Account (last private function)

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        loginBinding = ActivityLoginBinding.inflate(layoutInflater)
        val view = loginBinding.root
        setContentView(view)
    }
}
```

Εικόνα 3.20 Σύνδεση αρχείου xml με το αρχείο Kotlin - Login Activity

Στην περίπτωση που ο χρήστης δεν είχε αποσυνδεθεί την τελευταία φορά που χρησιμοποίησε την εφαρμογή, για να μην απαιτείται η επανειλημμένη εισαγωγή στοιχείων εισόδου, επιτρέπεται η απευθείας είσοδος στην κεντρική δραστηριότητα (Main Activity).

Αυτό επιτυγχάνεται με τη διαμόρφωση της συνάρτησης “override fun onStart()” που λειτουργεί αυτόματα κάθε φορά που ξεκινάει μια δραστηριότητα.

```
// If the user is already logged in, we will keep user signed until the user decide to sign out
// It's helpful, no sing up every time user open the application
// For bank applications is not good option. If I have logged in before,
override fun onStart() {
    super.onStart()
    //I will create a user object form the FirebaseAuth Class and assign it to the current user's data
    val user = auth.currentUser//this methode allow us to have information about the user

    if (user != null){ //if the user is not equal to null, that's mean the user is already logged in
        Toast.makeText(applicationContext, text: "Welcome Again", Toast.LENGTH_SHORT).show()
        //if user is already logged in, Go directly to the main Page
        val intent = Intent( packageContext: this, MainActivity :: class.java)
        startActivity(intent) //move to the next activity
        finish()//close this activity
    }
}
```

Εικόνα 3.21 Μεταφορά στην κεντρική δραστηριότητα

3.4.3.2 Είσοδος με τη χρήση κωδικού πρόσβασης

Για την αποφυγή σφαλμάτων, τα δεδομένα που εισάγει ο χρήστης ελέγχονται ως προς την πιθανότητα κενής εισόδου (user input = empty) (Εικόνα 3.22).

Στην περίπτωση που ικανοποιείται αυτή η συνθήκη καλείται η συνάρτηση signInUser μέσα στην οποία ελέγχονται τα δεδομένα και επιτρέπεται ή απορρίπτεται η είσοδος (Εικόνα 3.23). Σε κάθε περίπτωση εμφανίζεται μήνυμα στην οθόνη που ενημερώνει τον χρήστη για το αποτέλεσμα της προσπάθειας εισόδου.

```

loginBinding.buttonSignIn.setOnClickListener { 
    //get the email and password entered by the user
    val userEmail = loginBinding.editTextLoginEmail.text.toString()
    val userPassword = loginBinding.editTextLoginPassword.text.toString()
    //I need to check if the user input is null.
    // Without this if statement, in case the user input is empty the application will crush
    // In this if statement we provide a right input.
    if (userEmail == "" || userPassword == ""){
        loginBinding.buttonSignIn.isClickable = false
        Toast.makeText(applicationContext, text: "Please write the Email and the Password.",
            Toast.LENGTH_SHORT).show()
        loginBinding.editTextLoginEmail.setText("").toString()
        loginBinding.editTextLoginPassword.setText("").toString()
        loginBinding.buttonSignIn.isClickable = true
    }
    else{
        loginBinding.buttonSignIn.isClickable = true
        //I call the function for the SignIn
        signInUser(userEmail,userPassword)
    }
}
}

```

Εικόνα 3.22 Έλεγχος κενής εισόδου

```

//function for Sign in button // We need to create an object form the Firebase Auth class - in global area
fun signInUser(userEmail : String , userPassord : String){
    //it compares the previously created account information with the information currently entered.
    // If the information match each other, then it starts the login procedure
    // This is ready function made by android studio
    auth.signInWithEmailAndPassword(userEmail , userPassord).addOnCompleteListener { task →
        //Now the listener checks if the process has been completed
        if (task.isSuccessful){
            Toast.makeText(applicationContext, text: "Welcome to our team again!",Toast.LENGTH_SHORT).show() // Show the toast message
            val intent = Intent( packageContext: this@LoginActivity,MainActivity :: class.java) // go to the next activity (Main Activity)
            startActivity(intent) //start the new activity
            finish() //close this activity
        }
        else{
            Toast.makeText(applicationContext,task.exception?.localizedMessage,Toast.LENGTH_SHORT).show()
            //show me the reason why cannot be perform sign up (example: not valid type for email)
            //if the text is wrong delete all the content to have clear space
            loginBinding.editTextLoginEmail.setText("").toString()
            loginBinding.editTextLoginPassword.setText("").toString()
        }
    }
}
}
}

```

Εικόνα 3.23 Επαλήθευση στοιχείων χρήστη

3.4.3.3 Είσοδος με τη χρήση λογαριασμού Google

Μια άλλη επιλογή που επιτρέπει την πιο γρήγορη είσοδο χρήστη, χωρίς την επανειλημμένη εισαγωγή κωδικού, είναι η χρήση του λογαριασμού Google ως μέθοδο ταυτοποίησης. Για την υλοποίηση αυτής της επιλογής χρησιμοποιήθηκε έτοιμο κουμπί (που παρέχεται από τη βιβλιοθήκη 'com.google.android.gms:play-services-auth:20.1.0') και για αυτό το λόγο η διαμόρφωση του έγινε στο αρχείο Kotlin (Εικόνα 3.24) και όχι στο layout file όπως συμβαίνει στην περίπτωση όλων των υπόλοιπων UI στοιχείων.

```
// Configuration of Google Sign in Button!! This is because I used a ready button. I can always made button by my own
// The changes will be visible in the emulator not in the xml file SOS!!
// we access the textView defined inside this button (getChildAt function) //cast it to the TextView Class
val textOfGoogleButton = loginBinding.buttonGoogleSignIn.getChildAt(index: 0) as TextView
textOfGoogleButton.text = "Continue with Google" //here I can write the text that I want to appear in the Google log in button
textOfGoogleButton.setTextColor(Color.BLACK) //change the color of the text
textOfGoogleButton.textSize = 15F // F: value is floating

// **** functions for Google Sign in button ****
//register
registerActivityForGoogleSignIn()
```

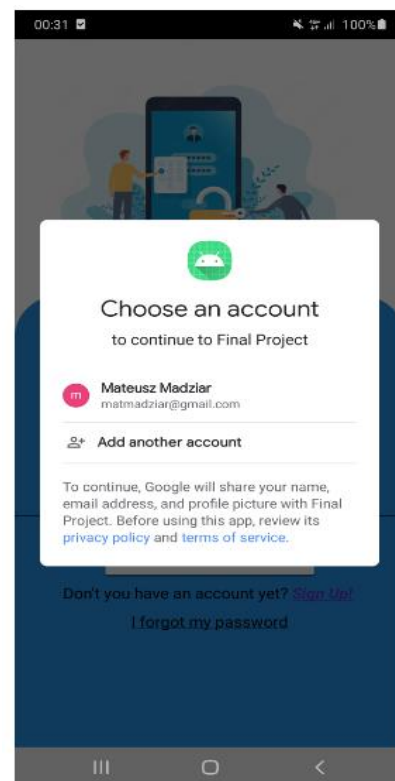
Εικόνα 3.24 Διαμόρφωση Google Sign-in Button – Layout

Στην περίπτωση που ο χρήστης δεν έχει συνδεθεί ξανά στην εφαρμογή μέσω λογαριασμού Google, είναι απαραίτητο να δημιουργηθεί νέος λογαριασμός, χρησιμοποιώντας έναν από τους λογαριασμούς που χρησιμοποιείται στο κινητό.

Με το πάτημα του «Google Sign-in Button» καλείται η συνάρτηση «**SignInGoogle()**» (Εικόνες 3.29, 3.30, 3.31) και δίνεται στον χρήστη η δυνατότητα επιλογής του λογαριασμού (σε περίπτωση που στο κινητό χρησιμοποιούνται παραπάνω από ένα ηλεκτρονικό ταχυδρομείο) (Εικόνα 3.25).

Η συνάρτηση «**registerActivityForGoogleSignIn()**» επιτρέπει τη συλλογή δεδομένων για τον λογαριασμό Google και τη σύνδεση με την πλατφόρμα της Firebase. Εκτελείται στο παρασκήνιο κάθε φορά που λειτουργεί η συγκεκριμένη δραστηριότητα (Login Activity).

Η λειτουργία της περιγράφεται λεπτομερώς στις εικόνες 3.26, 3.27 και 3.28.



Εικόνα 3.25 Επιλογή λογαριασμού Google


```
private fun registerActivityForGoogleSignIn(){ //to get the return data
    activityResultLauncher = registerForActivityResult(ActivityResultContracts.StartActivityForResult(),
        //2 parameters inside the brackets, the first one is standard,
        // the second parameter is a callback -so we get the return result with this parameter
        ActivityResultCallback { result → // here we can capture the result using the IT keyword
            val resultCode = result.resultCode //first I will get the result code and data
            val data = result.data

            if(resultCode == RESULT_OK && data != null){
                //this thread will get the data form Google in the background
                val task : Task<GoogleSignInAccount> = GoogleSignIn.getSignedInAccountFromIntent(data)
                // we will receive the data through this data object
                // then we will start the login process using the task object
                // we will write the login using new function
                firebaseSignInWithGoogle(task) //this function will be a function with one parameter and
                // it will take the task object as a parameter
            }
        }
    })
}
```

Εικόνα 3.26 Συνάρτηση registerActivityForGoogleSignIn()

```
private fun firebaseSignInWithGoogle (task : Task<GoogleSignInAccount> ){
    //if there's a problem while receiving data from the API, we take this error inside the catch block
    //For this reason we will make a "try" catch block here
    try {
        val account : GoogleSignInAccount = task.getResult(ApiException :: class.java)
        //now the signing is done, we can show the user a Toast message.
        Toast.makeText(applicationContext, text: "Welcome to our Team", Toast.LENGTH_SHORT).show()
        // the login page must be closed and home page must be open
        val intent = Intent( packageContext: this,MainActivity :: class.java)
        startActivity(intent)
        finish()
        // I call another function here
        firebaseGoogleAccount(account) // This function will take the account object as a parameter
    } catch (e : ApiException){ // if an error occurs, it will be an API exception error
        Toast.makeText(applicationContext, e.localizedMessage, Toast.LENGTH_SHORT).show()
    }
}
```

Εικόνα 3.27 Συνάρτηση firebaseSignInWithGoogle()

Σε περίπτωση απόρριψης πρόσβασης, εμφανίζεται στην οθόνη μήνυμα που ενημερώνει τον χρήστη για την αιτία του προβλήματος.

Σε αντίθετη περίπτωση ο χρήστης μεταφέρεται στην κεντρική δραστηριότητα (Main Activity), εμφανίζεται το ανάλογο μήνυμα και καλείται η συνάρτηση «firebaseGoogleAccount()» (Εικόνα 3.28).

```
private fun firebaseGoogleAccount(account: GoogleSignInAccount){
    //ID token has a unique code for each device// thanks to this code we can learn which device is logged in
    // we register the authenticated device of the person logging into the system with the token ID
    val authCredential = GoogleAuthProvider.getCredential(account.idToken, accessToken: null)
    //Now we will do the authentication process using the auth object we crated from firebaseAuth class
    auth.signInWithCredential(authCredential).addOnCompleteListener { task ->

        if (task.isSuccessful){
            // the important thing here is to retrieve the user's data if the login is done
            // val user = auth.currentUser
            // now I can access some date from the user using the user object
            //when the user logs in, it gives us the opportunity to access this data
            // in their Google account (name,email,photo URL)
        }
        else{
            //no need add else block at this moment
        }
    }
}
//***** End of Google Sign Button *****
```

Εικόνα 3.28 Συνάρτηση firebaseGoogleAccount()

Διαμόρφωση «κουμπιού» Εισόδου – ButtonGoogleSignIn:

```
// **** functions for Google Sign in button *****
// Configuration the google Sign in button
loginBinding.buttonGoogleSignIn.setOnClickListener { it.View()
    // I call the function for Sign with Google account
    signInGoogle()
}
```

Εικόνα 3.29 Λειτουργία Google Sign-in Button

```
// ***** functions for Google Sign in button *****
private fun signInGoogle(){
    val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    //Configuration Google standard Sign in methode with the builder methode
    .requestIdToken( serverClientId: "812867374367-6to8ipdp8lr1hpcv977p2alfhjgsfj6u.apps.googleusercontent.com")
    //This method specifies that an ID token for authenticated user is requested.
    // Requesting an ID token requires that the server client ID be specified.
    .requestEmail().build() // this will ask user for the email
    googleSignInClient = GoogleSignIn.getClient( activity: this,gso)
    signIn()
}
```

Εικόνα 3.30 Συνάρτηση signInGoogle()

```
// I will call it in google signInGoogle function
private fun signIn(){ // this wil start the intent sign in process
    val signInIntent : Intent = googleSignInClient.signInIntent
    // we can't run this intent with the start activity method. Instead we'll use the activity result launcher class,
    // because as a result of this intent we will receive some data from the user as a return
    // when we start this intent, the user will chose the Gmail address and we'll reach some data with this
    // I need to create an object form the activity result launcher class above****
    activityResultLauncher.launch(signInIntent)
}
```

Εικόνα 3.31 Συνάρτηση signIn()

3.4.4 Δημιουργία νέου λογαριασμού

3.4.4.1 Εγγραφή νέου χρήστη

Μέσω αυτής της δραστηριότητα παρέχεται η δυνατότητα δημιουργίας νέου λογαριασμού (Εικόνα 3.32). Το όνομα του αρχείου Kotlin είναι «SignUpActivity.kt».

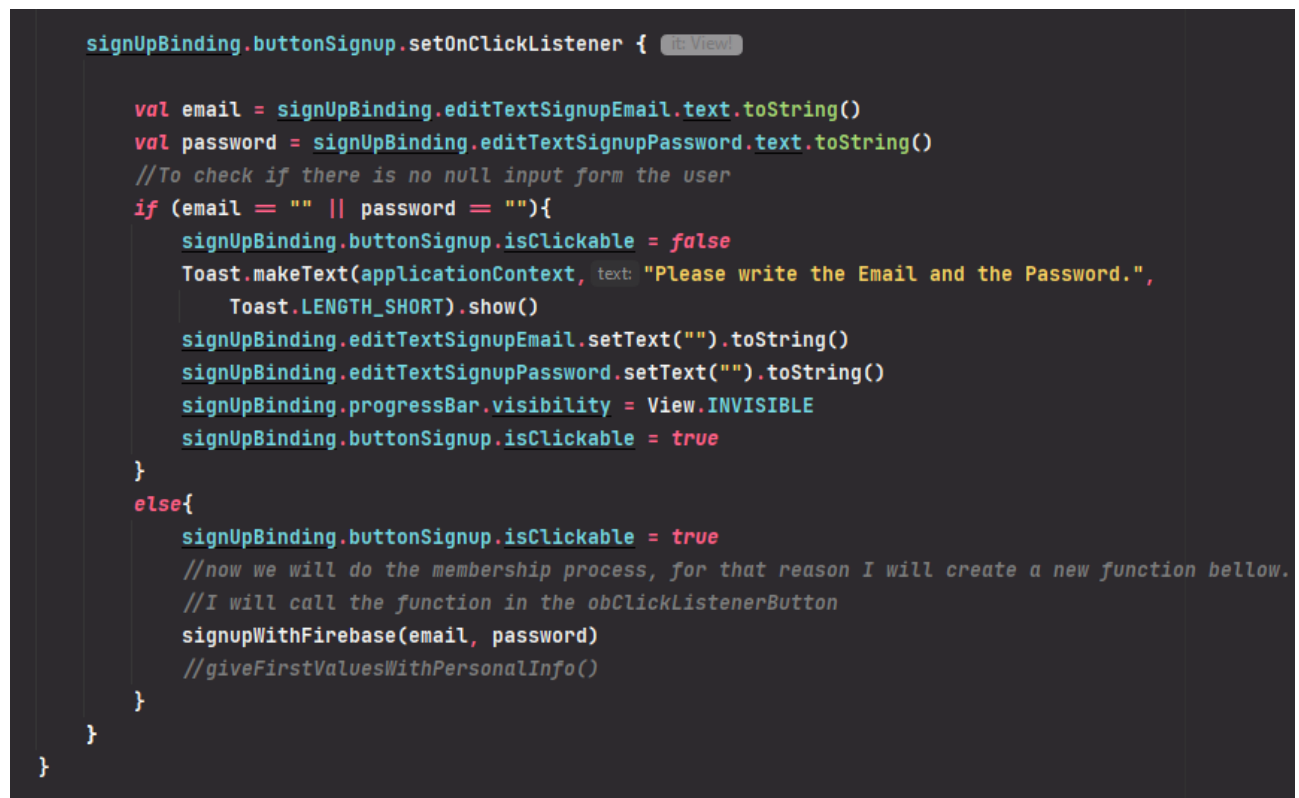
Τα στοιχεία που εισάγει ο χρήστης ελέγχονται με σκοπό την αποφυγή πιθανών σφαλμάτων στη λειτουργία της εφαρμογής (π.χ. κενή είσοδος) (Εικόνα 3.33). Διασφαλίζεται επίσης ο σωστός τύπος των δεδομένων εισόδου.

- Ο κωδικός πρόσβασης πρέπει να αποτελείται από τουλάχιστον 6 στοιχεία και να περιέχει τουλάχιστον ένα ειδικό χαρακτήρα.
- Στο πεδίο ηλεκτρονικού ταχυδρομείου τα δεδομένα που εισάγει ο χρήστης να είναι συγκεκριμένης μορφής (π.χ. «email@provider.com»)



Εικόνα 3.32 SignUp Activity

Στην περίπτωση που ο χρήστης προσπαθεί να δημιουργήσει καινούργιο λογαριασμό επιλέγοντας το ίδιο ηλεκτρονικό ταχυδρομείο που είχε επίλεξει στον παρελθόν το πρόγραμμα δεν επιτρέπεται την δημιουργία νέου λογαριασμού. Σε αυτήν, αλλά και σε κάθε άλλη περίπτωση που δεν είναι δυνατή η δημιουργία νέου λογαριασμού, ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα όπου αναφέρεται το πρόβλημα που παρουσιάστηκε.



Εικόνα 3.33 Διαμόρφωση του Sign Up Button

Με το πάτημα του κουμπιού «buttonSignUp» καλείται η συνάρτηση (Εικόνα 3.34), που είναι υπεύθυνη για τη δημιουργία νέου λογαριασμού στην πλατφόρμα της Firebase.

Επιπλέον, τερματίζει τη λειτουργία της συγκεκριμένης δραστηριότητας και μεταφέρει τον χρήστη στην επόμενη δραστηριότητα (SignUpAddAthleteInfoActivity), όπου αυτός καλείται να συμπληρώσει τα βασικά του στοιχεία και αναλύεται περαιτέρω στην υποενότητα 3.4.4.2 (Εικόνα 3.35).

```

fun signUpWithFirebase(email:String , password:String){

    //when I click the button, the progress bar should be visible (need to be activated - when the function is running)
    signUpBinding.progressBar.visibility = View.VISIBLE

    //The clickable property of the signup button must be false. I want the user can click the button ONLY ONE TIME.
    //Otherwise, membership may be created more than once (many creations for the same user)
    signUpBinding.buttonSignUp.isClickable = false
    /*Create user account:
    This function automatically creates an user account in Firebase and takes the email and password entered
    by the user as a parameter to create the account
    But it would make more sense to add a complete listener to this function to track the result of this operation
    I will also create a lambda representation with the task parameters. The listener check if this process is completed,
    we define the transactions to be performed in IF statement
    */
    auth.createUserWithEmailAndPassword(email, password).addOnCompleteListener { task →
        if (task.isSuccessful){
            //all the action we need when the account is created
            Toast.makeText(applicationContext, text: "Your account has been created",Toast.LENGTH_SHORT).show()
            //if the account has been created successfully, this page should be closed and the user should be directed to the section

            // send user to the SignUpAddAthleteInfoActivity to fill the basic information about the athlete.
            val intent = Intent( packageContext: this , SignUpAddAthleteInfoActivity ::class.java)
            startActivity(intent)
            finish() //close SignUp Activity. The transition to the login page will be made (parent activity)

            signUpBinding.progressBar.visibility = View.INVISIBLE //we need to make the progress bar invisible again
            signUpBinding.buttonSignUp.isClickable = true //The signupButton should be clickable again
        }
        else{
            //if cannot be opened, we can give the user the warning
            //shows the reason for the error (task.exception?.localizedMessage)
            Toast.makeText(applicationContext,task.exception?.localizedMessage,Toast.LENGTH_SHORT).show()
            signUpBinding.editTextSignUpEmail.setText("").toString()
            signUpBinding.editTextSignUpPassword.setText("").toString()
            signUpBinding.progressBar.visibility = View.INVISIBLE
            signUpBinding.buttonSignUp.isClickable = true
        }
    }
}
}

```

Εικόνα 3.34 Δημιουργία νέου λογαριασμού στη πλατφόρμα Firebase

3.4.4.2 Συμπλήρωση στοιχείων νέου χρήστη

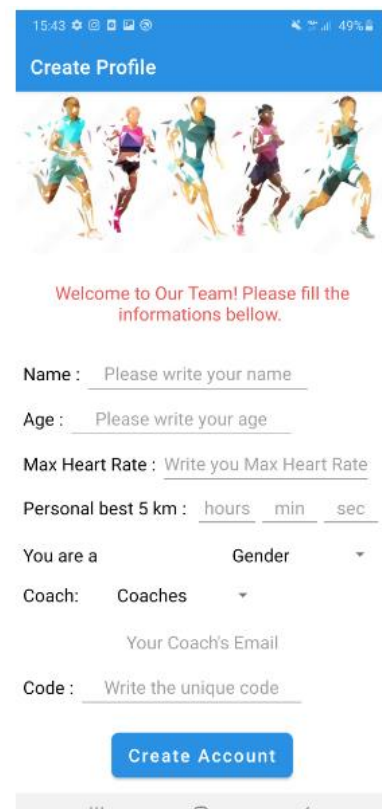
Το όνομα του αρχείου Kotlin αυτής της δραστηριότητας είναι «SignUpAddAthleteInfoActivity».

Τα δεδομένα που εισάγει ο χρήστης αποθηκεύονται στη βάση δεδομένων, στον υποφάκελο – userID, που έχει σαν όνομα τον ειδικό κωδικό κάθε χρήστη, του φακέλο «Athlete Info», όπως αναφέρεται στην υποενότητα 3.2 (Εικόνα 3.8, Εικόνα 3.9).

Με το πάτημα του κουμπιού «buttonCreateAccount» τα δεδομένα αποθηκεύονται στη βάση δεδομένων (Εικόνα 3.9). Για αυτό είναι υπεύθυνη η συνάρτηση της εικόνας 3.37.

Για να μπορέσει να γίνει μεταφορά και καταγραφή δεδομένων στη Firebase Realtime Database, ήταν απαραίτητη η κλήση συναρτήσεων που επιτρέπουν την αυθεντικοποίηση του χρήστη, αλλά και η δημιουργία μιας databaseReference (Εικόνα 3.36).

Στη συνέχεια η δραστηριότητα τερματίζεται και ο χρήστης μεταφέρεται στην κεντρική δραστηριότητα (MainActivity).



Εικόνα 3.35 Εισαγωγή βασικών πληροφοριών του αθλητή

```
val database = FirebaseDatabase.getInstance()
val auth: FirebaseAuth = FirebaseAuth.getInstance()
val databaseReferenceSignUp = database.reference.child(pathString: "Athlete Info")
val user = auth.currentUser
```

Εικόνα 3.36 Μεταφορά δεδομένων στη βάση δεδομένων, δημιουργία databaseReference

```
fun giveFirstValuesWithPersonalInfo() {
    databaseReferenceSignUp.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            user?.let {
                val userID = it.uid
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Name").setValue(name)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Age").setValue(age)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Max Heart Rate").setValue(maxHr)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Gender").setValue(genderOption)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Personal Best 5 km").child(pathString: "hours").setValue(pbHours)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Personal Best 5 km").child(pathString: "min").setValue(pbnMin)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Personal Best 5 km").child(pathString: "sec").setValue(pbSec)
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Coach Email").setValue(coachEmail).toString()
                databaseReferenceSignUp.child(userID).child(pathString: "Personal Info").child(pathString: "Coach Code").setValue(coachCode).toString()
            }
        }
    })
    override fun onCancelled(error: DatabaseError) {
        Toast.makeText(applicationContext, error.message, Toast.LENGTH_SHORT).show()
    }
}
```

Εικόνα 3.37 Συνάρτηση giveFirstValuesWithPersonalInfo()

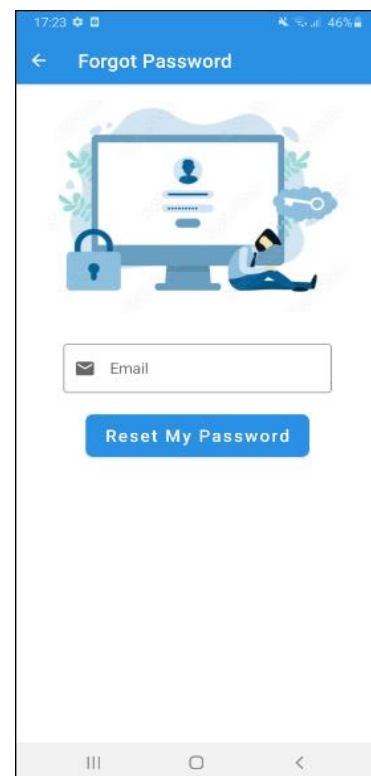
3.4.5 Επιλογή ανάκτησης κωδικού πρόσβασης

Το όνομα του αρχείου Kotlin αυτής της δραστηριότητας είναι «ForgotPasswordActivity».

Με το πάτημα του κουμπιού «Reset My Password» εκτελείτε ο κώδικας που περιγράφεται στην εικόνα 3.39.

Ο χρήστης λαμβάνει μήνυμα στο ηλεκτρονικό ταχυδρομείο που δήλωσε (Εικόνα 3.40). Πατώντας το ειδικό σύνδεσμο, του παρέχετε η δυνατότητα αλλαγής κωδικού πρόσβασης. Η δραστηριότητα τερματίζει και ο χρήστης μεταφέρεται στη δραστηριότητα εισόδου χρήστη (LoginActivity).

Όταν ολοκληρωθεί η διαδικασία αλλαγής κωδικού, ο χρήστης εισάγει τον καινούριο κωδικό και του επιτρέπεται η είσοδος στην εφαρμογή (Εικόνες 3.41, 3.42).

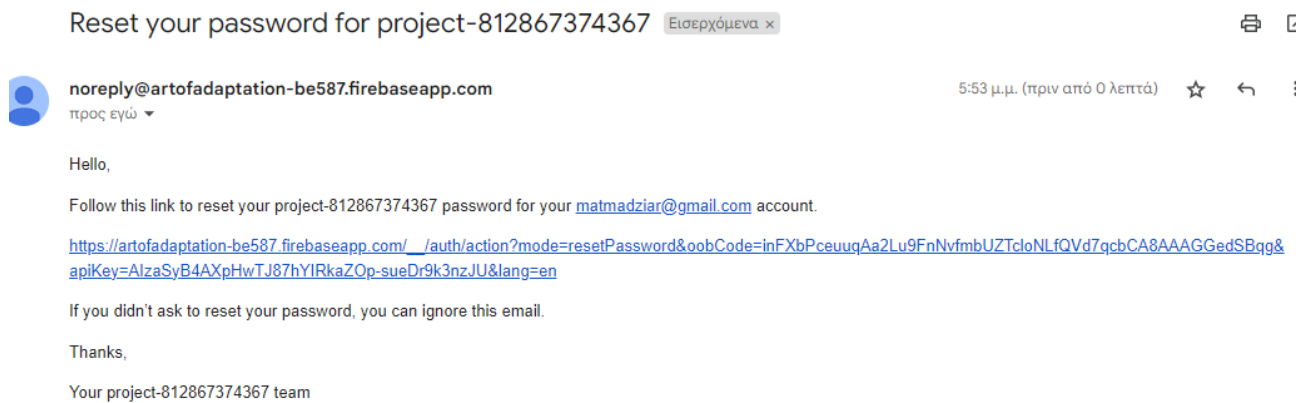


Εικόνα 3.38 Δραστηριότητα ανάκτησης κωδικού πρόσβασης

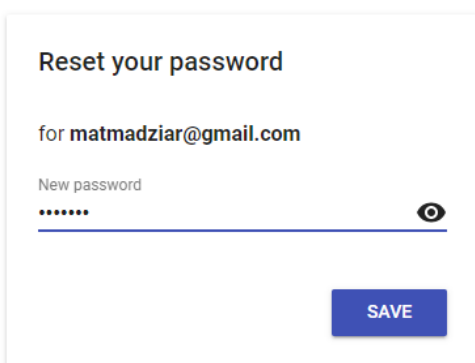
Το μήνυμα που αποστέλλετε στον χρήστη μαζί με τον ειδικό σύνδεσμο μπορεί να προσαρμοστεί, αλλάζοντας τις ρυθμίσεις που παρέχονται στην πλατφόρμα της Firebase. (Εικόνα 3.43)

```
forgotBinding.buttonReset.setOnClickListener {
    val userEmail = forgotBinding.editTextForgotEmail.text.toString() // Put user's input to the variable
    // We need to add a listener in case of a possible error
    // Ready function to send the reset password link to the user
    auth.sendPasswordResetEmail(userEmail).addOnCompleteListener { task ->
        if (task.isSuccessful){ //If the task is successful and the reset link is sent to the email address
            Toast.makeText(applicationContext
                , text: "We sent a password reset email to your email address"
                ,Toast.LENGTH_LONG).show()
            finish() // We will return to the login activity directly(parent activity)
        }
        else { //if the transaction is not succesful I define the actions needed to be performed here
            Toast.makeText(applicationContext,task.exception?.localizedMessage,Toast.LENGTH_SHORT).show()
        }
    }
}
}
```

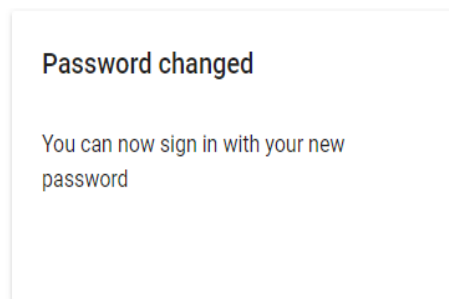
Εικόνα 3.39 Διαμόρφωση του κουμπιού ανάκτησης κωδικού



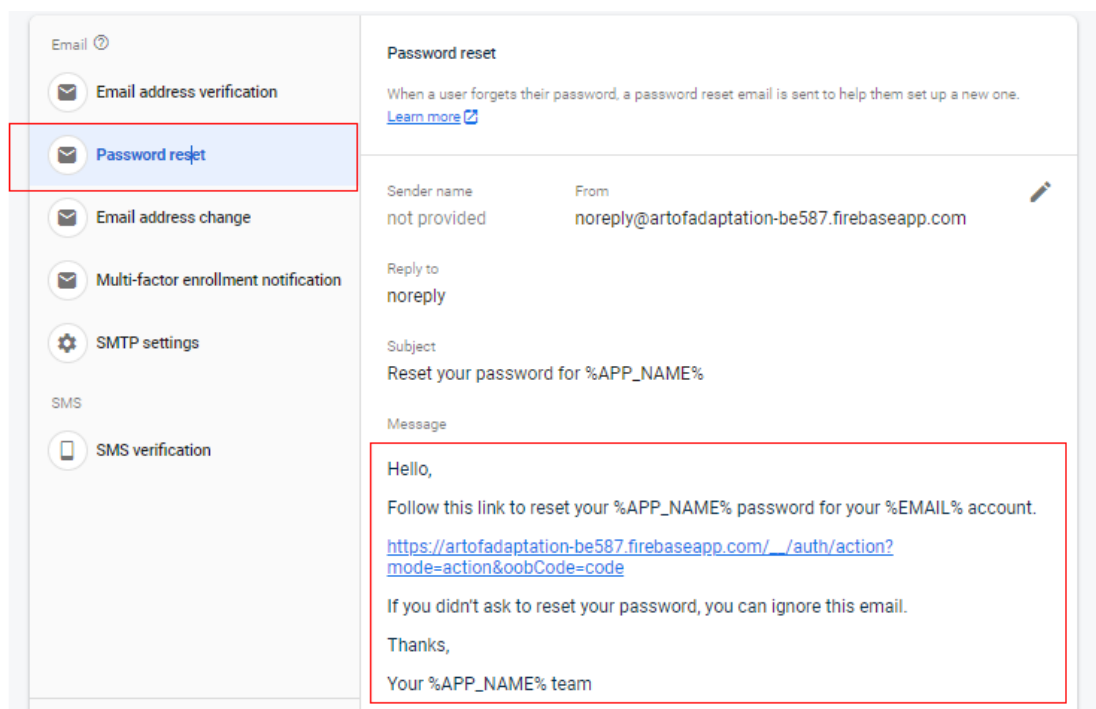
Εικόνα 3.40 Αλλαγή κωδικού πρόσβασης I



Εικόνα 3.41 Αλλαγή κωδικού πρόσβασης II



Εικόνα 3.42 Αλλαγή κωδικού πρόσβασης III



Εικόνα 3.43 Ρυθμίσεις μηνύματος για ανάκτηση κωδικού πρόσβασης

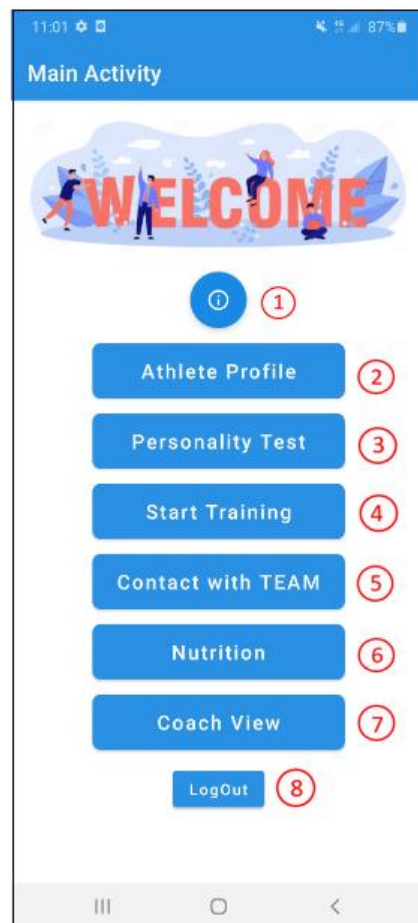
3.4.6 Κεντρική Δραστηριότητα

Θεωρείτε το σημείο αναφοράς της εφαρμογής. Σε αυτή περιέχονται όλες οι επιλογές που παρέχονται στον χρήστη.

Κάθε επιλογή (αρίθμηση εικόνας 3.44) μεταφέρει τον χρήστη σε διαφορετική δραστηριότητα.

- 1: Γενικές πληροφορίες (InformationsActivity)
- 2: Προφίλ του αθλητή (AthleteProfileActivity)
- 3: Ψυχολογικό τεστ (PersonalityTestActivity)
- 4: Εκκίνηση προπόνησης (StartTrainingActivity)
- 5: Επικοινωνία με την ομάδα υποστήριξης (ContactWithTeamActivity)
- 6: Διατροφή (NutritionActivity)
- 7: Επιλογές προπονητή (CoachViewActivity)
- 8: Αποσύνδεση / Αλλαγή χρήστη (LoginActivity)

Κάθε αθλητής έχει πρόσβαση σε όλες τις επιλογές, εκτός από την επιλογή «Coach View». Αυτή η επιλογή προορίζεται μόνο για τον προπονητή, προσφέροντας του, επιπλέον επιλογές. Ελέγχετε εάν ο λογαριασμός που συνδέθηκε είναι ένας από αυτούς που είναι δηλωμένος στη βάση δεδομένων στην ομάδα υποστήριξης και εάν τα στοιχεία συμφωνούν επιτρέπεται η πρόσβαση. Σε αντίθετη περίπτωση η επιλογή αυτή είναι απενεργοποιημένη (Εικόνα 3.45, Εικόνα 3.46).



Εικόνα 3.44 Κεντρική δραστηριότητα

```
private fun getUsersInfo() {
    val user = FirebaseAuth.currentUser
    user?.let {
        for (profile in it.providerData) {
            // Id of the provider (ex: google.com)
            val providerId = profile.providerId
            // UID specific to the provider
            val uid = profile.uid
            // Name, email address, and profile photo url
            // val name = profile.displayName
            email = profile.email.toString()
            // val photoUrl = profile.photoUrl
        }
    }
}
```

Εικόνα 3.45 Στοιχεία χρήστη

```
mainBinding.buttonCoachView.setOnClickListener {
    getUsersInfo()
    if (email == "head_coach@gmail.com" || email == "Gym_Coach@hotmail.com") {
        val intent = Intent(packageContext, this, CoachViewMainNewActivity::class.java)
        .putExtra(name: "userEmail", email.toString())
        startActivity(intent)
    } else {
        Toast.makeText(
            applicationContext,
            text: "You don't have access to this option",
            Toast.LENGTH_SHORT
        ).show()
    }
}
```

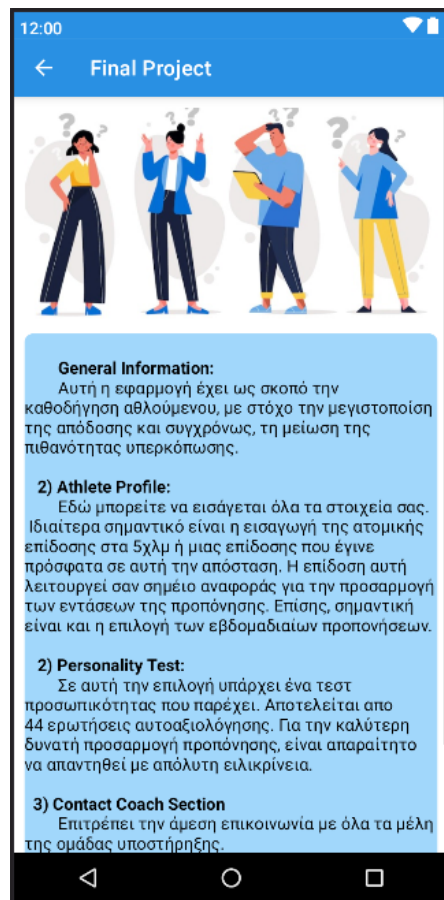
Εικόνα 3.46 Έλεγχος στοιχείων χρήστη

Παραπάνω πληροφορίες για τη δραστηριότητα του προπονητή βρίσκονται στην υποενότητα 3.4.13 αυτού του κεφαλαίου.

3.4.7 Γενικές πληροφορίες για τον τρόπο λειτουργίας της εφαρμογής

Σε αυτή τη δραστηριότητα αναφέρονται όλες οι απαραίτητες πληροφορίες που χρειάζεται ο χρήστης για να χρησιμοποιήσει με επιτυχία την εφαρμογή (Εικόνα 3.47). Περιγράφεται συνοπτικά η λειτουργία της κάθε επιλογής της κεντρικής δραστηριότητας.

Οι πληροφορίες είναι καταγεγραμμένες στον φάκελο «string.xml» (Εικόνα 3.48) γεγονός που επιτρέπει την ευκολότερη μελλοντική αξιοποίηση τους. Αυτός ο τρόπος παρέχει περισσότερες επιλογές στη διαμόρφωση του κειμένου ως προς το στυλιστικό κομμάτι.



Εικόνα 3.47 Δραστηριότητα πληροφοριών

```

<string name="GeneralInfo">
  <b>General Information:</b>
  Αυτή η εφαρμογή έχει ως σκοπό την καθοδήγηση αθλούμενου, με στόχο την μεγιστοποίηση
  της απόδοσης και συγχρόνως, τη μείωση της πιθανότητας υπερκόπωσης.
<b> 2) Athlete Profile: </b>
  Εδώ μπορείτε να εισάγετε όλα τα στοιχεία σας. \n
  Ιδιαίτερα σημαντικό είναι η εισαγωγή της ατομικής επίδοσης στα 5χλμ ή μιας επίδοσης που έγινε πρόσφατα σε αυτή την απόσταση.
  Η επίδοση αυτή λειτουργεί σαν σημείο αναφοράς για την προσαρμογή των εντάσεων της προπόνησης.
  Επίσης, σημαντική είναι και η επιλογή των εβδομαδιαίων προπονήσεων.
<b> 2) Personality Test: </b>
  Σε αυτή την επιλογή υπάρχει ένα τεστ προσωπικότητας που παρέχει. Αποτελείται από 44 ερωτήσεις αυτοαξιολόγησης.
  Για την καλύτερη δυνατή προσαρμογή προπόνησης, είναι απαραίτητο να απαντηθεί με απόλυτη ειλικρίνεια.
<b>3) Contact Coach Section </b>
  Επιτρέπει την άμεση επικοινωνία με όλα τα μέλη της ομάδας υποστήριξης.
</string>
    
```

Εικόνα 3.48 Γενικές πληροφορίες - αρχείο

3.4.8 Προφίλ Αθλούμενου

Σε αυτή τη δραστηριότητα ο χρήστης μπορεί να αλλάξει τα δεδομένα που καταχώρησε κατά την εγγραφή του στην εφαρμογή (Εικόνα 3.49).

Όταν ο χρήστης εισέρχεται σε αυτή τη δραστηριότητα, αυτόματα εμφανίζονται οι πληροφορίες που βρίσκονται στη βάση δεδομένων.

Αν υπάρξουν αλλαγές στα δεδομένα, με το πάτημα της επιλογής «Save» οι αλλαγές αποθηκεύονται απευθείας στη βάση δεδομένων και εμφανίζεται σχετικό μήνυμα.

- Η ηλικία, το φύλο είναι πληροφορίες που χρησιμοποιούνται στη δραστηριότητα «NutritionActivity» και βοηθάνε στον υπολογισμό των ημερήσιων θερμιδικών απαιτήσεων του αθλητή.
- Η μέγιστη καρδιακή συχνότητα είναι εξίσου σημαντική διότι επιτρέπει τον καθορισμό των ζωνών καρδιακών παλμών και εμφανίζονται στην δραστηριότητα «HRZones/PaceCalculatorActivity», αλλά και πριν την εκκίνηση της προπόνησης (για υπενθύμιση) στη δραστηριότητα «PaceCalculatorActivity».

The screenshot shows the 'Athlete Profile' screen. At the top, there's a blue header with a back arrow and the title 'Athlete Profile'. Below the header is a navigation bar with icons for Home, Profile, Calendar, and Messages. The main content area displays a grid of six circular profile icons. Below this, the following fields are visible:

- Name:** Mateusz
- Age:** 28
- Your gender is:** Options (dropdown), in DB: Male
- Max Heart Rate:** 202
- VO2 Max:** 63
- PB for 5km:** 0, 15, 30
- Training Per Week:** 5 (dropdown), Nr in DB: 5

At the bottom, there are two blue buttons: 'Back' and 'Save'.

Εικόνα 3.49 Προφίλ Αθλούμενου

- Η ατομική επίδοση κατέχει τον κύριο ρόλο στον καθορισμό των εντάσεων στις πιο απαιτητικές, αλλά και παράλληλα στις πιο σημαντικές προπονήσεις. Από τον συνολικό χρόνο υπολογίζονται οι ρυθμοί για την κάθε απόσταση για το κάθε προπονητικό κύκλο.
- Ο αριθμός των εβδομαδιαίων προπονήσεων είναι σημαντικός, διότι στη δραστηριότητα «TrainingPlanWeeks», όπου δίνεται η δυνατότητα επισκόπησης της κάθε εβδομάδας, εμφανίζονται μόνο οι προπονήσεις για τον επιλεγμένο αριθμό προπονήσεων. Αυτό συμβαίνει γιατί υπάρχουν διαφορετικά προπονητικά προγράμματα για αθλητή που επιλέγει να κάνει προπόνηση 3,4,5 ή 6 φορές την εβδομάδα.

Για την εμφάνιση δεδομένων χρησιμοποιείται η συνάρτηση της εικόνας 3.50, και καλείται στην αρχή του προγράμματος και εκτελείται κάθε φορά που ο χρήστης εισέρχεται σε αυτή τη δραστηριότητα.

Για την επιλογή του φύλου και του αριθμού εβδομαδιαίων προπονήσεων χρησιμοποιήθηκε το UI εργαλείο, το «spinner».

```
private fun RetrievePersonalInfoFromDataBase(){
    databaseReference.addValueEventListener(object : ValueEventListener {
        // This method constantly monitors the database life. If there's a change to the database, it instantly reflects to the application
        override fun onDataChange(snapshot: DataSnapshot) {
            user?.let { to FirebaseUser
                val userID = it.uid
                //take the information from the Firebase to the each element variable
                name = snapshot.child(userID).child( path: "Personal Info").child( path: "Name").value.toString()
                age = snapshot.child(userID).child( path: "Personal Info").child( path: "Age").value.toString()
                gender = snapshot.child(userID).child( path: "Personal Info").child( path: "Gender").value.toString()
                max_Hr = snapshot.child(userID).child( path: "Personal Info").child( path: "Max Heart Rate").value.toString()
                max_V02 = snapshot.child(userID).child( path: "Personal Info").child( path: "V02 Max").value.toString()
                pbHours = snapshot.child(userID).child( path: "Personal Info").child( path: "Personal Best 5 km").child( path: "hours").value.toString()
                pbnMin = snapshot.child(userID).child( path: "Personal Info").child( path: "Personal Best 5 km").child( path: "min").value.toString()
                pbSec = snapshot.child(userID).child( path: "Personal Info").child( path: "Personal Best 5 km").child( path: "sec").value.toString()
                numberOfTrainingsDB = snapshot.child(userID).child( path: "Personal Info").child( path: "Number of Trainings per Week").value.toString()
                athleteProfileBinding.editTextAthleteName.setText(name)
                athleteProfileBinding.editTextAthleteAge.setText(age)
                athleteProfileBinding.textViewGenderInfo.setText("in DB: $gender")
                athleteProfileBinding.editTextMaxHeartRate.setText(max_Hr)
                athleteProfileBinding.editTextMaxV02.setText(max_V02)
                athleteProfileBinding.editTextAthletePersonalBestHours.setText(pbHours)
                athleteProfileBinding.editTextAthletePersonalBestMinutes.setText(pbnMin)
                athleteProfileBinding.editTextAthletePersonalBestSeconds.setText(pbSec)
                athleteProfileBinding.textViewNrDataBase.setText("Nr in DB:$numberOfTrainingsDB")
                athleteProfileBinding.progressBarAthleteProfile.visibility = View.INVISIBLE
                athleteProfileBinding.LinearLayoutName.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutAge.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutGender.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutHeartRate.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutV02Max.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutPersonalBest.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutButtonsAthleteProfile.visibility = View.VISIBLE
                athleteProfileBinding.LinearLayoutNumberOfTrainings.visibility = View.VISIBLE } }
            override fun onCancelled(error: DatabaseError) {
                Toast.makeText(applicationContext, error.message, Toast.LENGTH_SHORT).show() // show the Toast message if an error occurs
            }
        }
    })
}
```

Εικόνα 3.50 Εμφάνιση δεδομένων αθλητή

Για την αποθήκευση δεδομένων στη βάση δεδομένων καλείται η παρακάτω συνάρτηση (Εικόνα 3.51)

```
fun sendInfoToDatabase(){
    user?.let { //I check if the user object is not null
        val userID = it.uid
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Name").setValue(athleteName)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Age").setValue(athleteAge)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Gender").setValue(athleteGender)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Max Heart Rate").setValue(athleteMaxHr)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "V02 Max").setValue(athleteMaxV02)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Personal Best 5 km").child( pathString: "hours").setValue(athletePbHours)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Personal Best 5 km").child( pathString: "min").setValue(athletePbnMin)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Personal Best 5 km").child( pathString: "sec").setValue(athletePbSec)
        infoRef.child(userID).child( pathString: "Personal Info").child( pathString: "Number of Trainings per Week").setValue(numberOfTrainings)
        Toast.makeText(applicationContext, text: "Profile informations are added to database", Toast.LENGTH_SHORT).show()
    }
}
```

Εικόνα 3.51 Αποθήκευση αλλαγών στο προφίλ αθλητή

Επιλέγοντας τη συγκεκριμένη δραστηριότητα, ο χρήστης καλείται να συμπληρώσει το τεστ αυτοαξιολόγησης που προσδιορίζει τα βασικά στοιχεία της προσωπικότητας του.

Η συνάρτηση «PersonalityTestLogic()» (εικόνα 3.53) εκτελείται αυτόματα όταν ο χρήστης εισέρχεται στη δραστηριότητα, αλλά και κάθε φορά που επιλέγει την επιλογή «next» για να μεταβεί στη επόμενη ερώτηση. Με τη χρήση της εμφανίζονται οι ερωτήσεις που βρίσκονται στη βάση δεδομένων.

Με τη βοήθεια την εντολής «snapshot.childrenCount» είναι γνωστός ο αριθμός των ερωτήσεων και σε περίπτωση που μελλοντικά αλλάξει αυτός, το πρόγραμμα θα λειτουργεί κανονικά χωρίς επιπλέον αλλαγές στον κώδικα λειτουργίας.



Εικόνα 3.52 Ψυχολογικό Προφίλ

```
private fun PersonalityTestLogic(){
    databaseReference.addValueEventListener(object : ValueEventListener{
        // This method constantly monitors the database life.
        // If there's a change to the database, it instantly reflects to the application
        override fun onDataChange(snapshot: DataSnapshot) {
            // if there will be changes in the number of the questions, the code is going to work without further changes
            questionCount = snapshot.childrenCount.toInt() // We know the total number of the questions
            if (questionNumber <= questionCount) {
                //take the question part of the first question to the question variable
                question = snapshot.child(questionNumber.toString()).child( path: "q").value.toString()
                answerA = snapshot.child(questionNumber.toString()).child( path: "a").value.toString()
                answerB = snapshot.child(questionNumber.toString()).child( path: "b").value.toString()
                answerC = snapshot.child(questionNumber.toString()).child( path: "c").value.toString()
                answerD = snapshot.child(questionNumber.toString()).child( path: "d").value.toString()
                answerE = snapshot.child(questionNumber.toString()).child( path: "e").value.toString()
                //Print the question in the text View
                personalityTestBinding.textViewQuestion.text = question // .setText(question) - is correct also ??
                // Print the answer options in the text View (Layout)
                personalityTestBinding.textViewA.text = answerA
                personalityTestBinding.textViewB.text = answerB
                personalityTestBinding.textViewC.text = answerC
                personalityTestBinding.textViewD.text = answerD
                personalityTestBinding.textViewE.text = answerE
                // When the data are retrieved, the progress bar should be invisible
                personalityTestBinding.progressBar5.visibility = View.INVISIBLE
                // The linearLayouts should be visible
                personalityTestBinding.linearLayoutQuestion.visibility = View.VISIBLE
                personalityTestBinding.linearLayoutButtons.visibility = View.VISIBLE
            }else{
                //When all the questions are finished, Show a Toast message
                Toast.makeText(applicationContext, text: "You answered all the questions", Toast.LENGTH_SHORT).show()
            }
            questionNumber ++ // will be increasing until the number of question will be over
        }
        override fun onCancelled(error: DatabaseError) {
            Toast.makeText(applicationContext, error.message, Toast.LENGTH_SHORT).show()
        }
    })
}
```

Εικόνα 3.53 Συνάρτηση PersonalityTestLogic()

Σχεδίαση και Ανάπτυξη Συστήματος Επίβλεψης και Καθοδήγησης της Απόδοσης Αθλούμενων

Ο υπολογισμός των επιμέρους παραμέτρων παρουσιάζεται στις εικόνες 3.54, 3.55, 3.56, 3.57, 3.58 και υπολογίζεται με βάση τις πληροφορίες που βρίσκονται στην εικόνα 1.4 του 1^{ου} κεφαλαίου. Για την απλοποίηση του κώδικα, οι ερωτήσεις στη βάση δεδομένων αποθηκεύτηκαν με διαφορετική σειρά με τέτοιο τρόπο, ώστε οι ερωτήσεις να αφορούν (είναι αποθηκευμένα σε ανάλογη σειρά) τα επίπεδα εξωστρέφειας (extraversion), συμφωνησιμότητας (agreeableness), ευσυνειδησίας (conscientiousness), νευρωτισμού (neuroticism) και ανοιχτότητας (openness). Η βαθμολογία υπολογίζεται με βάση τον πίνακα 1-3 του 1^{ου} κεφαλαίου. Στις ερωτήσεις όπου απαιτείται η αντιστροφή της βαθμολογίας, οι πιθανές απαντήσεις αλλάζουν σειρά στη βάση δεδομένων.

```
//The 4 possible answers
personalityTestBinding.textViewA.setOnClickListener {
    userAnswer = "a"
    userAnswerPoints ++
    personalityTestBinding.textViewA.setBackgroundColor(Color.YELLOW)
    disableClickableOfOptions()
    if (questionNumber <= 9){
        extroversionPoints ++
    }
    else if (questionNumber >9 && questionNumber <= 18){
        agreeablenessPoints ++
    }
    else if (questionNumber > 18 && questionNumber <= 27){
        conscientiousnessPoints ++
    }
    else if (questionNumber >27 && questionNumber <= 35){
        neuroticismPoints ++
    }
    else if (questionNumber >35 && questionNumber <= 45){
        OpennessToExperiencePoints ++
    }
}
```

Εικόνα 3.54 Υπολογισμός παραμέτρων I

```
personalityTestBinding.textViewB.setOnClickListener {
    userAnswer = "b"
    userAnswerPoints = userAnswerPoints + 2.0
    personalityTestBinding.textViewB.setBackgroundColor(Color.YELLOW)
    disableClickableOfOptions()
    if (questionNumber <= 9){
        extroversionPoints = extroversionPoints + 2.0
    }
    else if (questionNumber >9 && questionNumber <= 18){
        agreeablenessPoints = agreeablenessPoints + 2.0
    }
    else if (questionNumber > 18 && questionNumber <= 27){
        conscientiousnessPoints = conscientiousnessPoints + 2.0
    }
    else if (questionNumber >27 && questionNumber <= 35){
        neuroticismPoints = neuroticismPoints + 2.0
    }
    else if (questionNumber >35 && questionNumber <= 45){
        OpennessToExperiencePoints = OpennessToExperiencePoints + 2.0
    }
}
```

Εικόνα 3.55 Υπολογισμός παραμέτρων II

```
personalityTestBinding.textViewC.setOnClickListener {
    userAnswer = "c"
    userAnswerPoints = userAnswerPoints + 3.0
    personalityTestBinding.textViewC.setBackgroundColor(Color.YELLOW)
    disableClickableOfOptions()
    if (questionNumber <= 9){
        extroversionPoints = extroversionPoints + 3.0
    }
    else if (questionNumber >9 && questionNumber <= 18){
        agreeablenessPoints = agreeablenessPoints + 3.0
    }
    else if (questionNumber > 18 && questionNumber <= 27){
        conscientiousnessPoints = conscientiousnessPoints + 3.0
    }
    else if (questionNumber >27 && questionNumber <= 35){
        neuroticismPoints = neuroticismPoints + 3.0
    }
    else if (questionNumber >35 && questionNumber <= 45){
        OpennessToExperiencePoints = OpennessToExperiencePoints + 3.0
    }
}
```

Εικόνα 3.56 Υπολογισμός παραμέτρων III

```
personalityTestBinding.textViewD.setOnClickListener {
    userAnswer = "d"
    userAnswerPoints = userAnswerPoints + 4.0
    personalityTestBinding.textViewD.setBackgroundColor(Color.YELLOW)
    disableClickableOfOptions()
    if (questionNumber <= 9){
        extroversionPoints = extroversionPoints + 4.0
    }
    else if (questionNumber >9 && questionNumber <= 18){
        agreeablenessPoints = agreeablenessPoints + 4.0
    }
    else if (questionNumber > 18 && questionNumber <= 27){
        conscientiousnessPoints = conscientiousnessPoints + 4.0
    }
    else if (questionNumber >27 && questionNumber <= 35){
        neuroticismPoints = neuroticismPoints + 4.0
    }
    else if (questionNumber >35 && questionNumber <= 45){
        OpennessToExperiencePoints = OpennessToExperiencePoints + 4.0
    }
}
```

Εικόνα 3.57 Υπολογισμός παραμέτρων IV

```

personalityTestBinding.textViewE.setOnClickListener {
    userAnswer = "g"
    userAnswerPoints = userAnswerPoints + 5.0
    personalityTestBinding.textViewE.setBackgroundColor(Color.YELLOW)
    disableClickableOfOptions()
    if (questionNumber ≤ 9){
        extroversionPoints = extroversionPoints + 5.0
    }
    else if (questionNumber >9 && questionNumber ≤ 18){
        agreeablenessPoints = agreeablenessPoints + 5.0
    }
    else if (questionNumber > 18 && questionNumber ≤27){
        conscientiousnessPoints = conscientiousnessPoints + 5.0
    }
    else if (questionNumber>27 && questionNumber ≤35){
        neuroticismPoints = neuroticismPoints + 5.0
    }
    else if(questionNumber>35 && questionNumber≤45){
        OpennessToExperiencePoints = OpennessToExperiencePoints + 5.0
    }
}
}

```

Εικόνα 3.58 Υπολογισμός παραμέτρων V

Αποθήκευση αποτελεσμάτων με την επιλογή του κουμπιού «Finish» (Εικόνα 3.59).

```

fun saveResultsToDatabase(){
    user?.let{
        val userID = it.uid
        PersonalityScoreRef.child(userID).child( pathString: "Personal Info").child( pathString: "Result Psychological Test Results")
            .child( pathString: "Extroversion").setValue(extroversionPoints)
        PersonalityScoreRef.child(userID).child( pathString: "Personal Info").child( pathString: "Result Psychological Test Results")
            .child( pathString: "Agreeableness").setValue(agreeablenessPoints)
        PersonalityScoreRef.child(userID).child( pathString: "Personal Info").child( pathString: "Result Psychological Test Results")
            .child( pathString: "Conscientiousness").setValue(conscientiousnessPoints)
        PersonalityScoreRef.child(userID).child( pathString: "Personal Info").child( pathString: "Result Psychological Test Results")
            .child( pathString: "Neuroticism").setValue(neuroticismPoints)
        PersonalityScoreRef.child(userID).child( pathString: "Personal Info").child( pathString: "Result Psychological Test Results")
            .child( pathString: "Openness to Experience").setValue(OpennessToExperiencePoints)
        Toast.makeText(applicationContext, text: "Score sent to database successfully", Toast.LENGTH_SHORT).show()
        val intent = Intent( packageContext: this , MainActivity ::class.java)
        startActivity(intent)
        finish()
    }
}
}

```

Εικόνα 3.59 Αποθήκευση αποτελέσματος ψυχολογικού προφίλ

Τα αποτελέσματα αποθηκεύονται στις βασικές πληροφορίες του αθλητή όπως αυτό φαίνεται στη εικόνα 3.9.

3.4.10.1 Περίληψη

Σε αυτή τη δραστηριότητα παρέχονται στον χρήστη επιλογές που παρουσιάζονται παρακάτω.

Με την επιλογή της κάθε επιλογής, ο αθλητής μεταφέρεται σε διαφορετική δραστηριότητα (Εικόνα 3.60).

Επιλογές:

- 1: Δραστηριότητα στην οποία παρουσιάζεται το εβδομαδιαίο πρόγραμμα προπόνησης. (TrainingPlanWeeksActivity)
- 2: Δραστηριότητα που παρέχονται πληροφορίες για τις ζώνες καρδιακών παλμών, αλλά και οι ρυθμοί που πρέπει να ακολουθεί ο κάθε αθλητής. (HRZonesPaceCalculatorActivity)
- 3: Δραστηριότητα η οποία περιέχει όλες τις προπονήσεις που έχει πραγματοποιήσει ο αθλητής. (MyTrainingsActivity)
- 4: Δραστηριότητα που μεταφέρει τον αθλητή στις ερωτήσεις πριν από την προπόνηση. (QuestionsBeforeTrainingActivity)
- 5: Καιρικές συνθήκες (μελλοντική επέκταση – WeatherConditionsActivity).
- 6: Επιστροφή την κεντρική δραστηριότητα (MainActivity),



Εικόνα 3.60 Start Training Activity

Κάθε φορά που ο αθλητής πραγματοποιεί την καθημερινή του προπόνηση, καλείται να επιλέξει την επιλογή «Start Training». Τότε ξεκινάει η διαδικασία καταγραφής δεδομένων που μία από τις βασικότερες λειτουργίες της εφαρμογής που υλοποιήθηκε.

Σε αυτή τη δραστηριότητα ο χρήστης έχει τη δυνατότητα να δει όλες τις προπονήσεις και για τις 12 προπονητικές εβδομάδες. Σε περίπτωση που ο προπονητής κάνει αλλαγές στο πρόγραμμα, θα εμφανιστούν αυτές σε πραγματικό χρόνο.

Η συγκεκριμένη δραστηριότητα έχει άμεση σχέση με τη δραστηριότητα «AthleteProfileActivity». Ανάλογα τον αριθμό εβδομαδιαίων προπονήσεων που έχει επιλέξει ο χρήστης, εμφανίζεται το ανάλογο προπονητικό πρόγραμμα.

Στην Εικόνα 3.61 παρουσιάζεται το εβδομαδιαίο πρόγραμμα της δεύτερης εβδομάδας για 3 προπονήσεις την εβδομάδα.

Στην Εικόνα 3.62 παρουσιάζεται το αντίστοιχο εβδομαδιαίο πρόγραμμα της δεύτερης εβδομάδας, αλλά για 5 προπονήσεις εβδομαδιαίως.

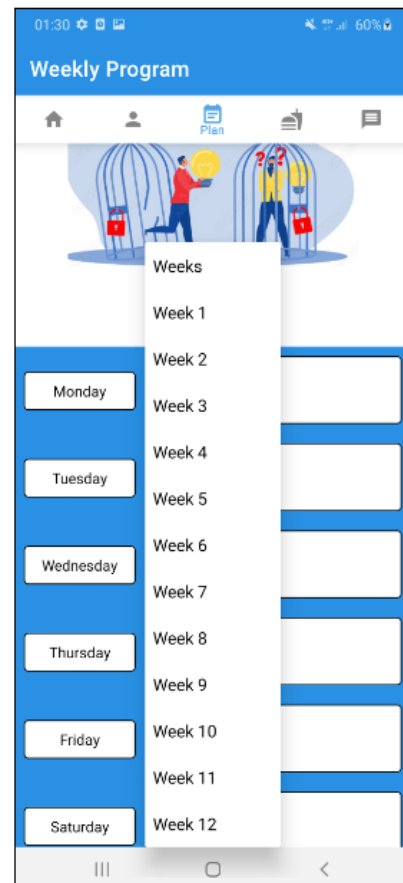
Στην Εικόνα 3.63 παρουσιάζονται οι επιλογές που έχει ο αθλητής.



Εικόνα 3.61 Εβδομάδες Προπόνησης I



Εικόνα 3.62 Εβδομάδες Προπόνησης II



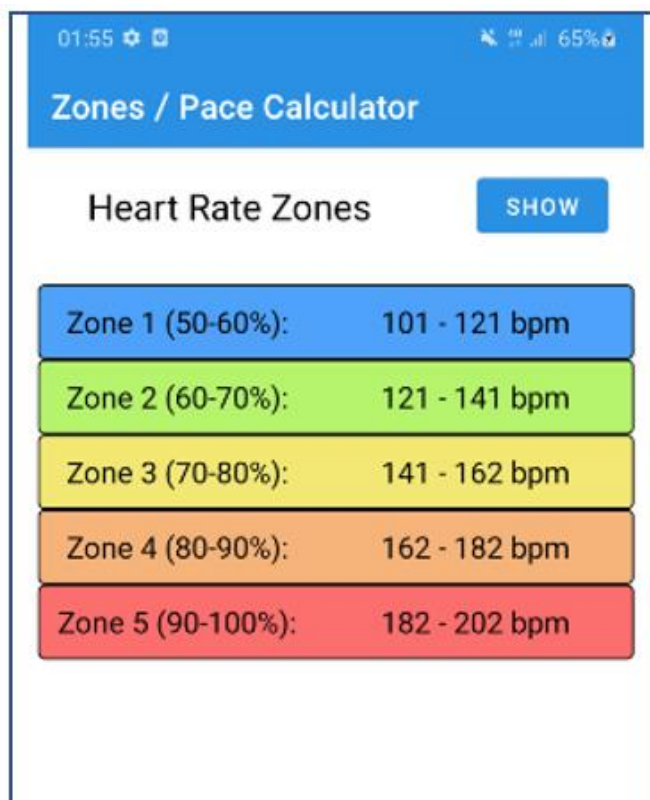
Εικόνα 3.63 Εβδομάδες Προπόνησης III

Αυτή η δραστηριότητα έχει ως σκοπό την ενημέρωση του αθλητή για το εύρος της κάθε ζώνης καρδιακών παλμών. Είναι σημαντικές διότι επιτρέπουν την πραγματοποίηση προπόνησης ελαχιστοποιώντας το ενεργειακό κόστος και την λειτουργία του μεταβολικού συστήματος που απαιτείται κάθε φορά.

Οι ζώνες καρδιακών παλμών υπολογίζονται με βάση τα δεδομένα της δραστηριότητας «AthleteProfileActivity».

Στην Εικόνα 3.64 παρουσιάζονται οι ζώνες καρδιακών παλμών για αθλητή με μέγιστη καρδιακή συχνότητα 202 παλμούς/δευτερόλεπτο.

Επιπλέον, ο αθλητής ενημερώνεται για τους ρυθμούς προπόνησης που πρέπει να ακολουθεί σε κάθε προπονητική μονάδα. Στην Εικόνα 3.65 παρουσιάζονται οι ρυθμοί για αποστάσεις από 200 μέτρα, μέχρι και 10 χιλιόμετρα.



Εικόνα 3.64 Ζώνες καρδιακών παλμών



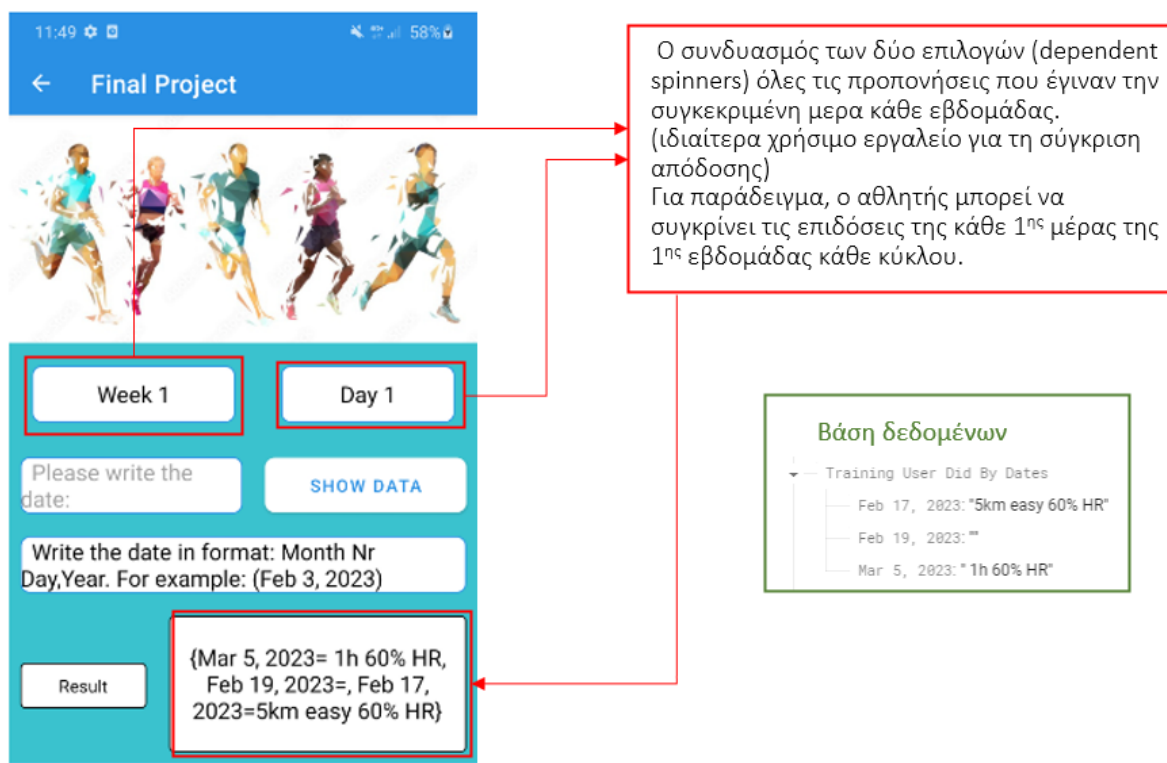
Εικόνα 3.65 Ρυθμοί προπόνησης

* Αξίζει να σημειωθεί, ότι ο χρόνος των 10 χλμ. που φαίνεται στην εικόνα 3.65 υπολογίστηκε σαν διπλάσιο του χρόνου, που απαιτεί ο αθλητής να καλύψει της απόσταση των 5χλμ. Σε πραγματικές συνθήκες, για μια σωστή προσέγγιση ισχύει ότι: (ο χρόνος των 5χλμ x 2) + 1 λεπτό και 20 δευτερόλεπτα, το οποίο όμως δεν είναι απόλυτο και αλλάζει ανάλογα με το επίπεδο του αθλητής και την ειδική προπόνηση που έχει κάνει για το συγκεκριμένο αγώνισμα.

Μέσω αυτής της δραστηριότητας παρέχεται στον αθλητή η δυνατότητα επίβλεψης όλων των προπονήσεων που έχουν καταγράψει για κάθε μέρα, κάθε εβδομάδα του εκάστοτε προπονητικού κύκλου. Επιπλέον, έχει δυνατότητα σύγκρισης των αποτελεσμάτων με προηγούμενους προπονητικούς κύκλους.

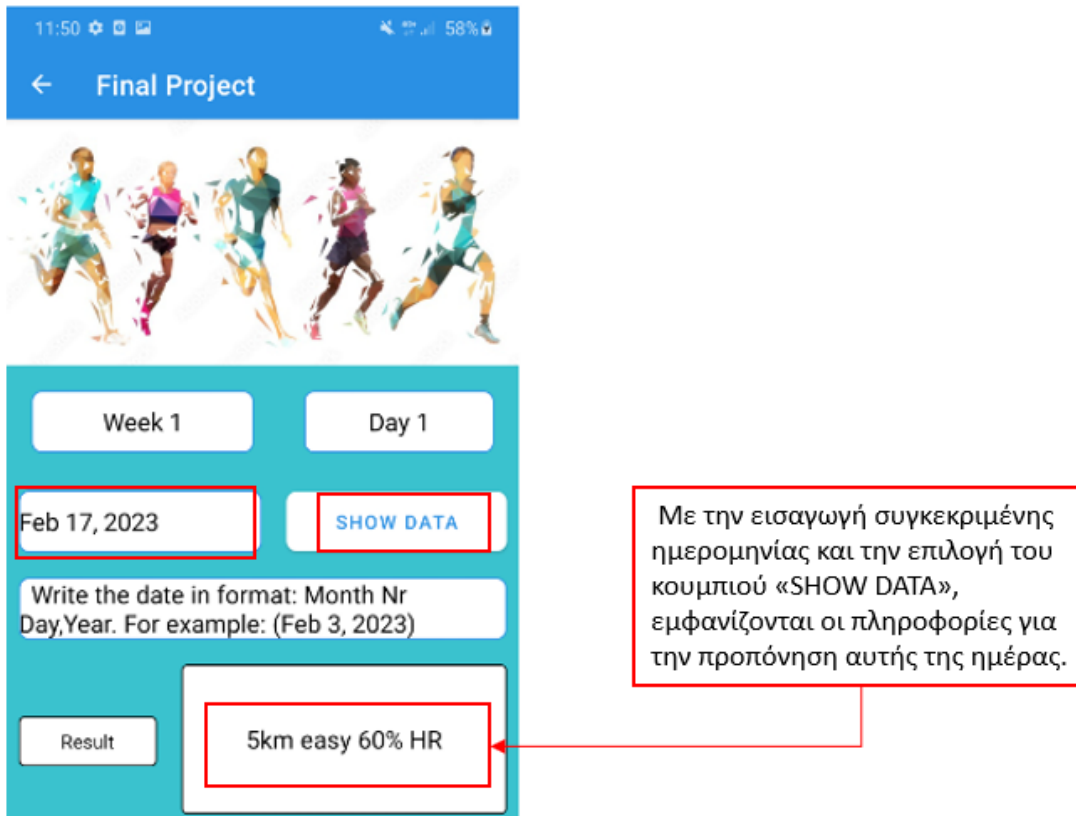
Ενσωματώθηκαν 3 επιλογές αναζήτησης.

- Υπάρχει δυνατότητα επιλογής εβδομάδας και μέρας του προπονητικού κύκλου (εικόνα 3.66). Αυτή η επιλογή ενδεικνύεται κυρίως για αθλητές οι οποίοι χρησιμοποιούν την εφαρμογή μετά το πέρας του 1^{ου} προπονητικού κύκλου (12 εβδομάδες). Η συγκεκριμένη επιλογή επιτρέπει την απευθείας σύγκριση απόδοσης μεταξύ 2 ή και παραπάνω προπονητικών κύκλων.

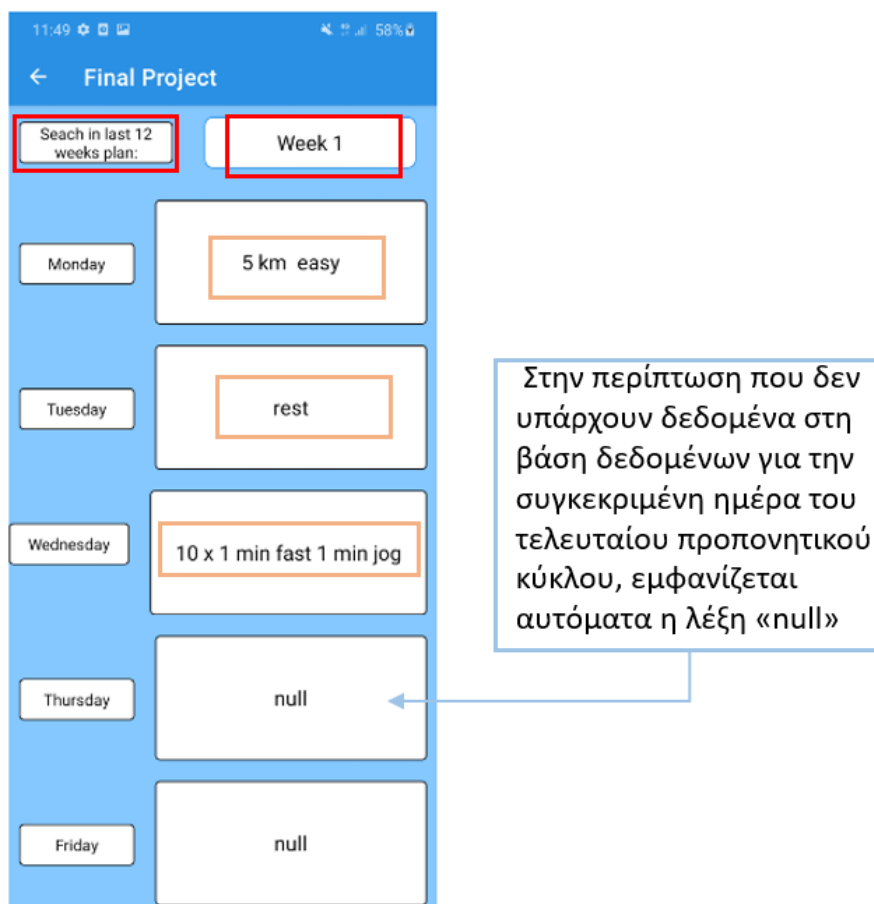


Εικόνα 3.66 Επιλογή αναζήτησης δεδομένων I

- Με την εισαγωγή συγκεκριμένης ημερομηνίας (Εικόνα 3.67).
- Εμφάνιση των προπονήσεων του τελευταίου προπονητικού κύκλου με την επιλογή συγκεκριμένης εβδομάδας (Εικόνα 3.68).



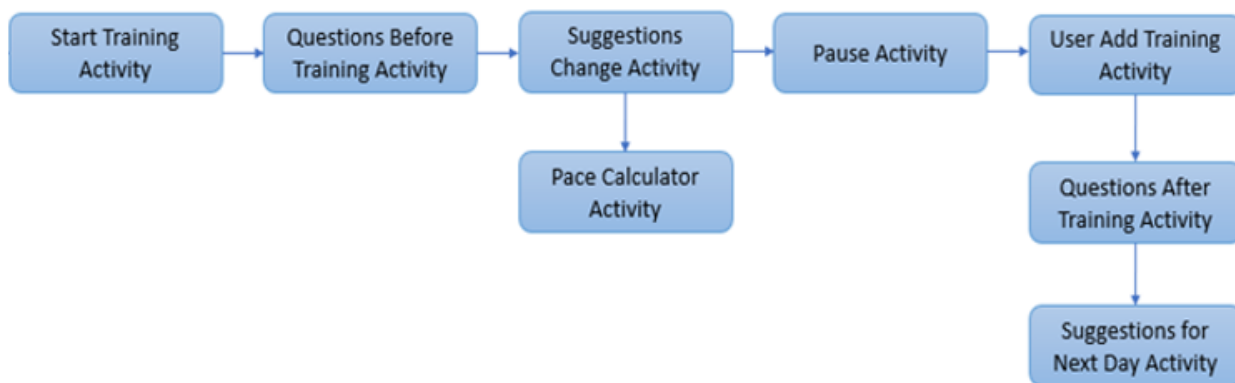
Εικόνα 3.67 Επιλογή αναζήτησης δεδομένων II



Εικόνα 3.68 Επιλογή αναζήτησης δεδομένων III

3.4.10.5.1 **Περίληψη**

Πατώντας την επιλογή «Start Training Today» ο χρήστης καλείται να ακολουθήσει μια διαδικασία που έχει στόχο την κατάλληλη προσαρμογή της ημερήσιας επιβάρυνσης και την παροχή συμβουλών. Σε αυτή την υποενότητα (3.4.10.5) περιγράφεται η λειτουργία της κάθε επιμέρους δραστηριότητας (Εικόνα 3.69).



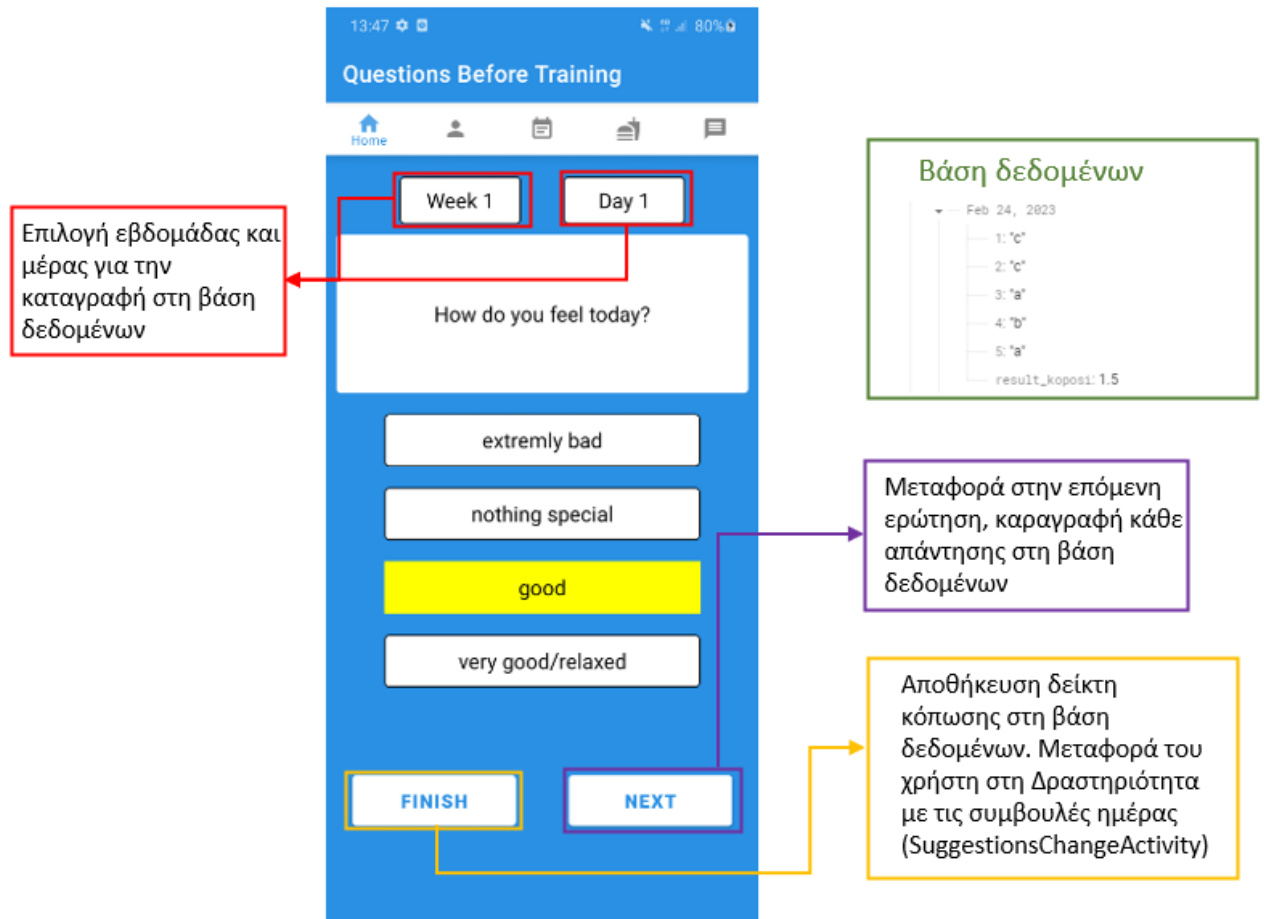
Εικόνα 3.69 Διάγραμμα Start Training Activity

3.4.10.5.2 **Ερωτήσεις πριν την προπόνηση**

Το ερωτηματολόγιο αποτελείται από 5 ερωτήσεις και οι ερωτήσεις βρίσκονται στη βάση δεδομένων, γεγονός που διευκολύνει την πιθανή μελλοντική ένταξη περισσότερων ερωτήσεων.

Εάν αλλάξει ο αριθμός τους δε χρειάζεται να γίνουν περαιτέρω αλλαγές στον κώδικα λειτουργίας. Αυτό επιτυγχάνεται με τη χρήση της συνάρτησης «snapshot.childrenCount» με παρόμοιο τρόπο όπως υλοποιήθηκε το ψυχολογικό τεστ που περιγράφεται στην ενότητα 3.4.9 (εικόνα 3.53). Η δομή της δραστηριότητας παρουσιάζεται στην εικόνα 3.70.

Είναι κρίσιμης σημασίας η επιλογή εβδομάδας και ημέρας, μιας και είναι απαραίτητα στοιχεία τόσο στην ανάκτηση συγκεκριμένων πληροφοριών από τη βάση δεδομένων, όσο και για τη σωστή καταγραφή πληροφοριών που εισάγει ο χρήστης σε όλη τη διάρκεια καταγραφής ημερήσιας δραστηριότητας.



Εικόνα 3.70 Ερωτήσεις πριν την προπόνηση

3.4.10.5.3 Συμβουλές για άμεσες αλλαγές

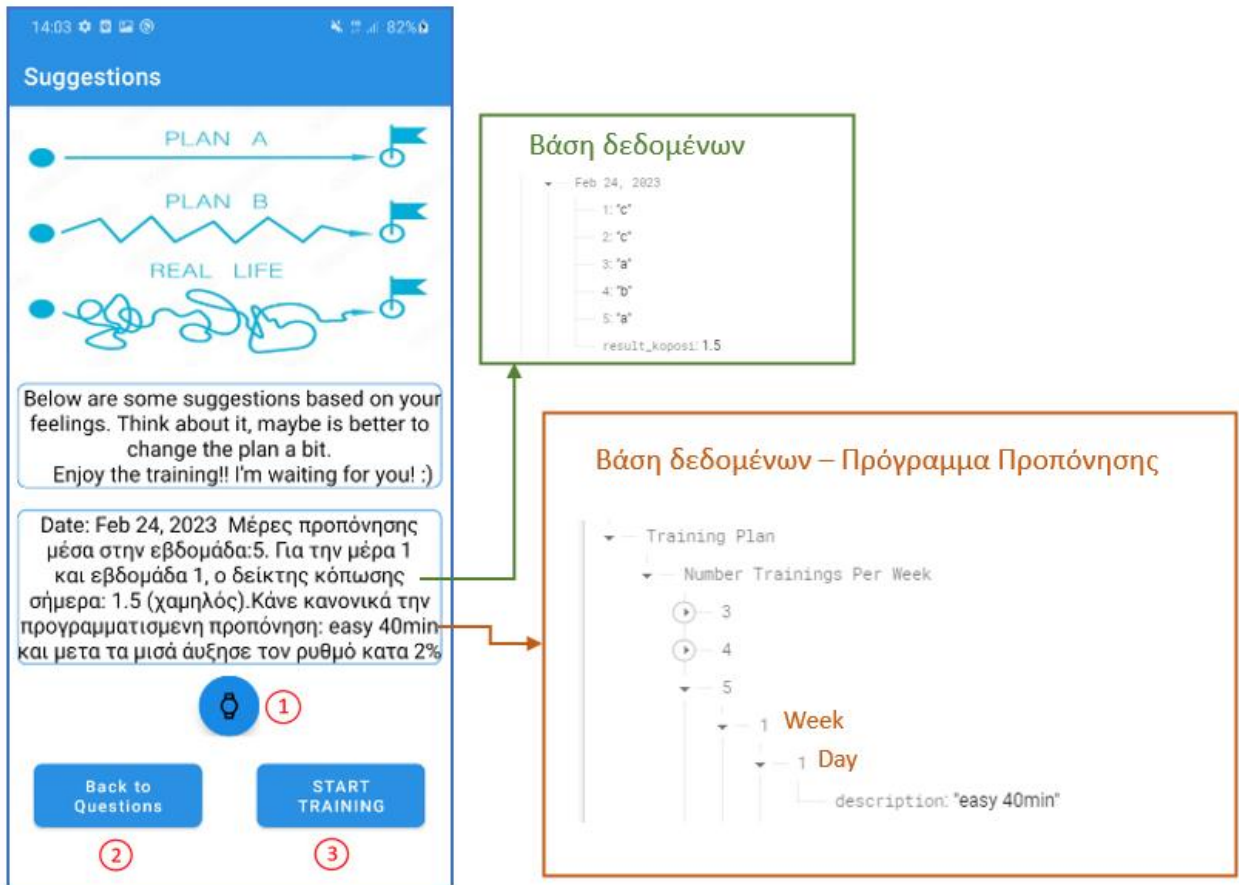
Σε αυτή τη δραστηριότητα προτείνονται οι αλλαγές που μπορεί να κάνει ο χρήστης στην προπόνηση της ημέρας (Εικόνα 3.71).

Συγκεντρώνονται όλα τα στοιχεία που αφορούν:

- Την ημέρα του προπονητικού κύκλου
- Την εβδομάδα του προπονητικού κύκλου
- Την ημερομηνία, η οποία καταγράφεται αυτόματα από το κινητό
- Τον δείκτη κόπωσης, που υπολογίζεται με γνώμονα τους παράγοντες που επηρεάζουν την απόδοση του αθλητή (αναφέρονται λεπτομερώς στην ενότητα 1.2).
- Το πρόγραμμα που είναι αποθηκευμένο στη βάση δεδομένων για την συγκεκριμένη μέρα.
- Τη συμβουλή για πιθανή τροποποίηση του προγράμματος

Επιπλέον, παρέχονται οι παρακάτω επιλογές:

- 1: Εμφάνιση δραστηριότητας με τις ζώνες καρδιακών παλμών και τους προτεινόμενους ρυθμούς (ShowPaceActivity, Εικόνες 3.64, 3.65)
- 2: Επιστροφή στην προηγούμενη δραστηριότητα (QuestionsBeforeTrainingActivity).
- 3: Μεταφορά του χρήστη στην επόμενη δραστηριότητα (PauseActivity).

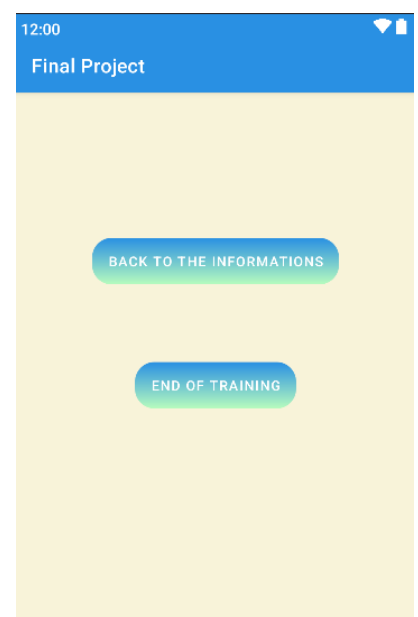


Εικόνα 3.71 Συμβουλές πριν την προπόνηση

3.4.10.5.4 *Pause Activity*

Αυτή η δραστηριότητα εμφανίζεται στην οθόνη κατά τη διάρκεια που ο αθλητής εκτελεί την προπόνηση (Εικόνα 3.72). Δεν υπάρχει κάποιο ιδιαίτερο ενδιαφέρον όσο αφορά τη λειτουργία της.

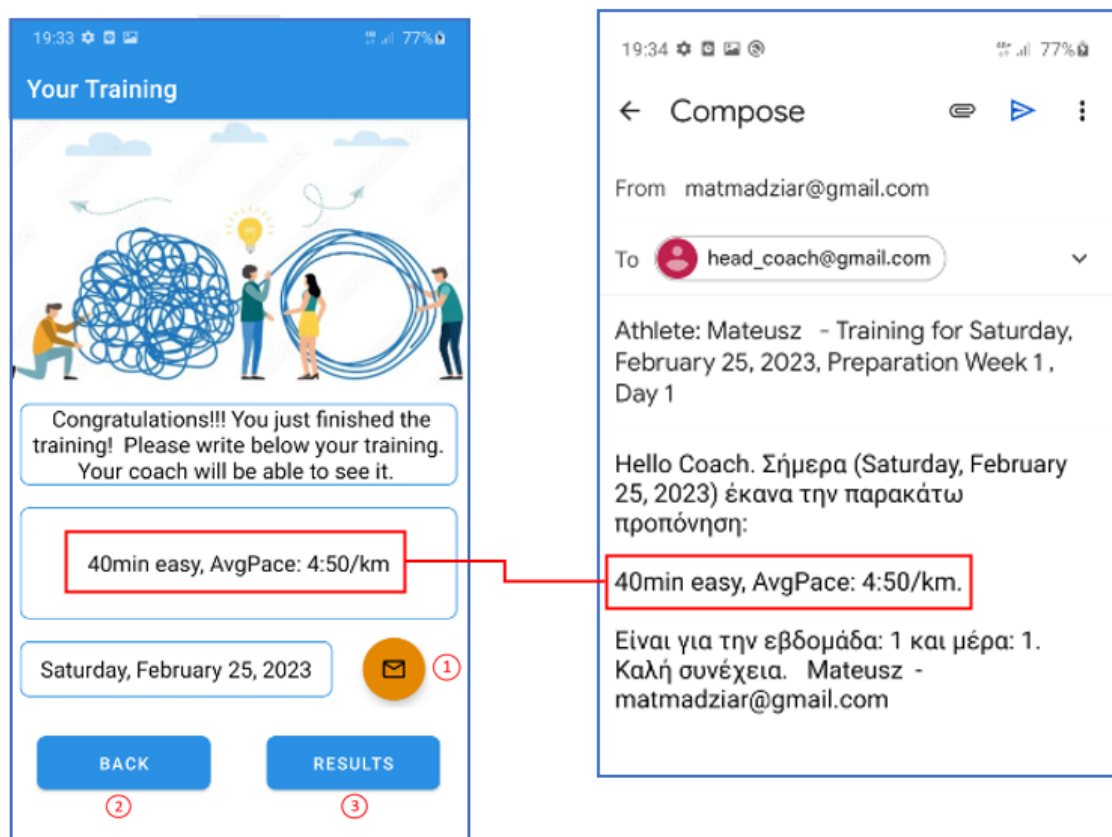
Σε περίπτωση που ο χρήστης επιθυμεί να δει ξανά τις συμβουλές, έχει τη δυνατότητα επιστροφής στην προηγούμενη δραστηριότητα επιλέγοντας «BACK TO THE INFORMATIONS». Όταν ολοκληρωθεί η προπόνηση, με την επιλογή «END OF TRAINING», ο χρήστης μεταφέρεται στην επόμενη δραστηριότητα («UserAddTrainingActivity»).



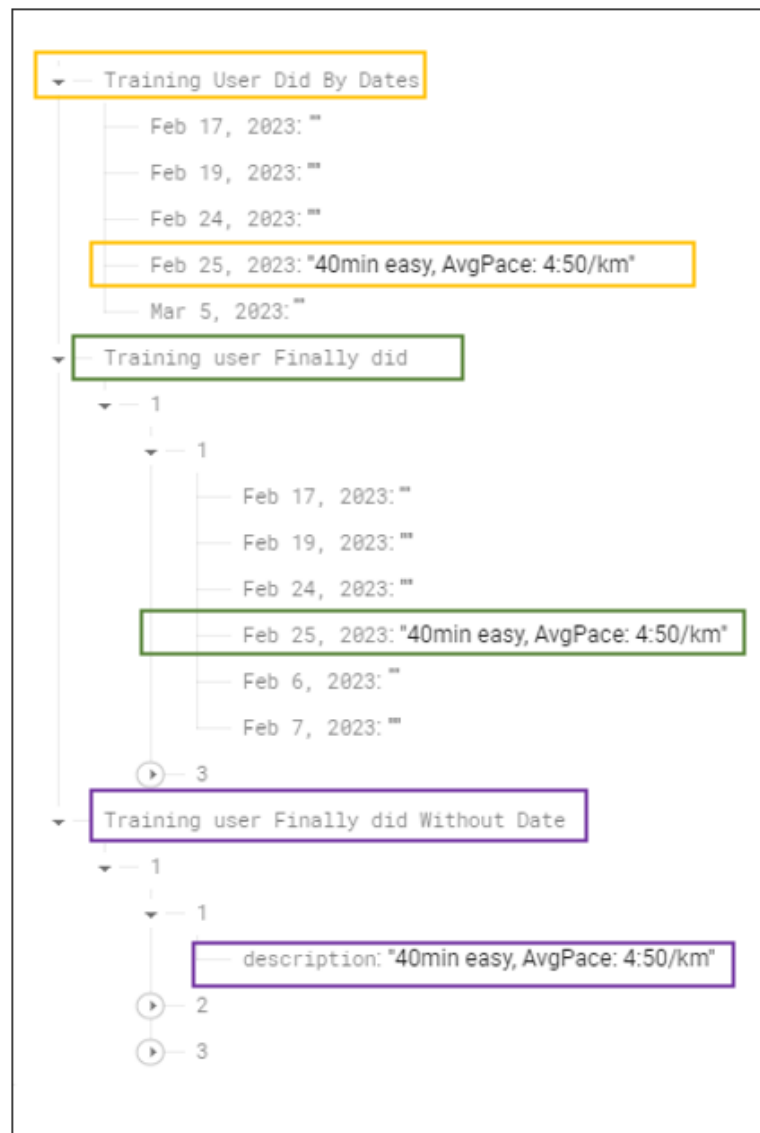
Εικόνα 3.72 PauseActivity

Αυτή η δραστηριότητα επιτρέπει την αποθήκευση δεδομένων, αλλά και την αποστολή τους στον προπονητή.

- Επιλογή «1»: Τα δεδομένα αποθηκεύονται στη βάση δεδομένων σε 3 διαφορετικούς φακέλους (Εικόνα 3.74) και ο χρήστης μεταφέρεται στην επόμενη δραστηριότητα. Αυτό επιτρέπει την αναζήτηση προπονήσεων με διαφορετικούς τρόπους στις δραστηριότητες «MyTrainingsActivity», αλλά και «EachUserProgressActivity» που περιγράφονται λεπτομερώς στις ενότητες 3.4.10.4 και 3.4.13.3 αντίστοιχα.
- Επιλογή «2»: Επιτρέπει στον αθλητή την άμεση αποστολή πληροφοριών στον προπονητή του για την προπόνηση που εκτέλεσε, μέσω εφαρμογής ηλεκτρονικού ταχυδρομείου της επιλογής του, μαζί με όλες τις απαραίτητες πληροφορίες (αυτοματοποιημένη συμπλήρωση όλων των στοιχείων), όπως η ημερομηνία, μέρα και εβδομάδα του προπονητικού κύκλου, αλλά και τα στοιχεία του αθλητή (Εικόνα 3.73).
- Επιλογή «3»: Επιστροφή στην προηγούμενη δραστηριότητα (PauseActivity).



Εικόνα 3.73 Καταγραφή δραστηριότητας και αποστολή δεδομένων



Εικόνα 3.74 Αποθήκευση προπονήσεων στη βάση δεδομένων

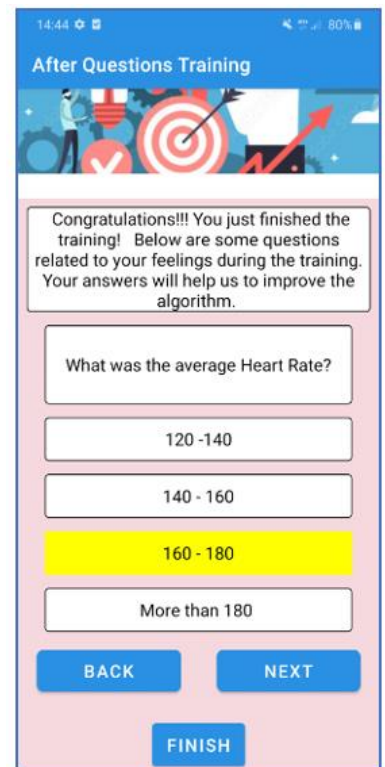
Τη στιγμή που ο χρήστης μεταφέρεται στην εφαρμογή που επιτρέπει την αποστολή μηνύματος με τη χρήση ηλεκτρονικού ταχυδρομείου, τα δεδομένα παραμένουν στο πεδίο εισαγωγής και όταν επιστρέψει ξανά, δεν απαιτείται η νέα καταγραφή τους. Περισσότερες πληροφορίες για τον κώδικα λειτουργίας που επιτρέπει την αποστολή δεδομένων με τη χρήση ηλεκτρονικού ταχυδρομείου αναφέρονται στην ενότητα 3.4.11.2.

Μετά την ολοκλήρωση της καταγραφής της ημερήσιας προπόνησης, ο αθλητής μεταφέρεται σε αυτή τη δραστηριότητα, όπου καλείται να απαντήσει σε ερωτήσεις που αφορούν την αίσθηση του κατά την άσκηση, αλλά και βασικούς παράγοντες (μετρήσεις από ζώνη καρδιακών παλμών), απο τους οποίους προκύπτει ο δείκτης κόπωσης (εικόνα 3.75).

Δεν απαιτείται από τον χρήστη να επιλέξει ξανά την εβδομάδα και μέρα την οποία αφορούν οι απαντήσεις, διότι τα δεδομένα αυτά βρίσκονται στη βάση δεδομένων.

Με την επιλογή «NEXT» η κάθε απάντηση αποθηκεύεται στη βάση δεδομένων και εμφανίζεται η επόμενη ερώτηση.

Με την επιλογή «FINISH» προκύπτει μια εκτίμηση για τον δείκτη κόπωσης και ο χρήστης μεταφέρεται στην επόμενη δραστηριότητα (SuggestionForNextDayActivity).



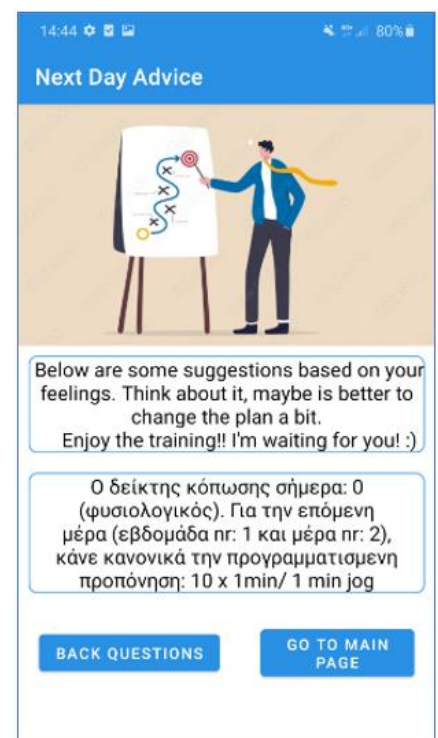
Εικόνα 3.75 Ερωτήσεις μετά την προπόνηση

3.4.10.5.7 **Συμβουλές για την επόμενη προπόνηση**

Με βάση τις απαντήσεις της προηγούμενης δραστηριότητας προτείνονται κάποιες αλλαγές στο προπονητικό πρόγραμμα για τις επόμενες ημέρες. Εμφανίζεται αυτόματα η προπόνηση της επόμενης μέρας. Διασφαλίζεται η ομαλή λειτουργία τις εφαρμογής μετά την ολοκλήρωση της τελευταίας ημέρας της 12^{ης} εβδομάδας. Ο χρήστης καλείται να ακολουθήσει ξανά τον προπονητικό κύκλο από την αρχή (εικόνα 3.76).

Με την επιλογή «Go to Main Page» ολοκληρώνεται η διαδικασία καταγραφής προπόνησης και ο αθλητής μεταφέρεται στην κεντρική δραστηριότητα (MainActivity).

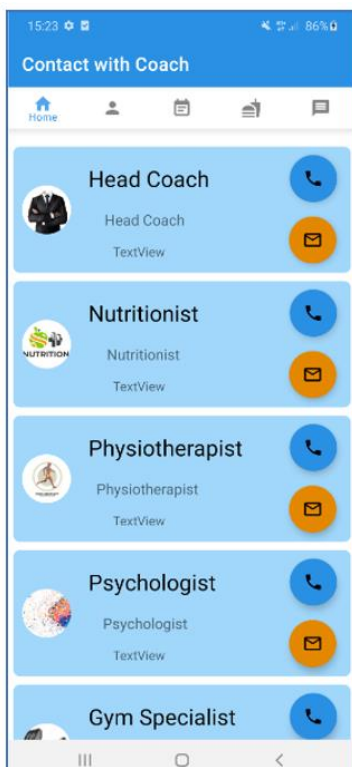
Οι συμβουλές αποθηκεύονται στη βάση δεδομένων και μπορούν να χρησιμοποιηθούν στη μελλοντική μελέτη της αξιοπιστίας της εφαρμογής.



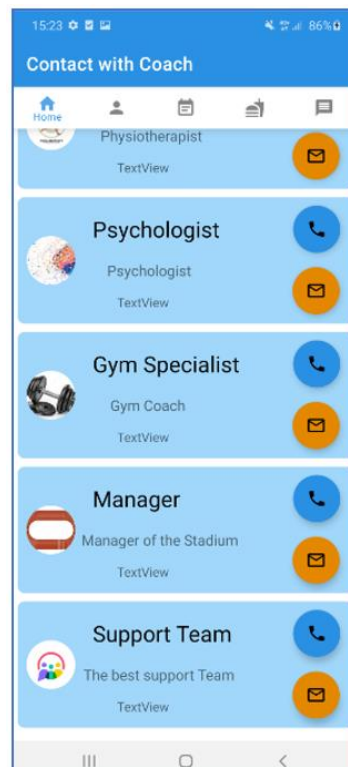
Εικόνα 3.76 Συμβουλές για την επόμενη ημέρα

3.4.11.1 **Περίληψη**

Ιδιαίτερα σημαντική είναι η εξασφάλιση της άμεσης επικοινωνίας με όλα τα μέλη της ομάδας υποστήριξης. Τα δεδομένα για το κάθε μέλος βρίσκονται αποθηκευμένα στη βάση δεδομένων (Εικόνα 3.5), γεγονός που επιτρέπει την ανανέωση τους σε πραγματικό χρόνο, εάν υπάρξει τέτοια ανάγκη (Εικόνες 3.77, 3.78). Ο αθλητής έχει την επιλογή να επικοινωνήσει με την ομάδα του τηλεφωνικά (Ενότητα 3.4.11.2), αλλά και με τη χρήση οποιασδήποτε εφαρμογής που υποστηρίζει την αποστολή μηνυμάτων με τη χρήση ηλεκτρονικού ταχυδρομείου (Ενότητα 3.4.11.3).



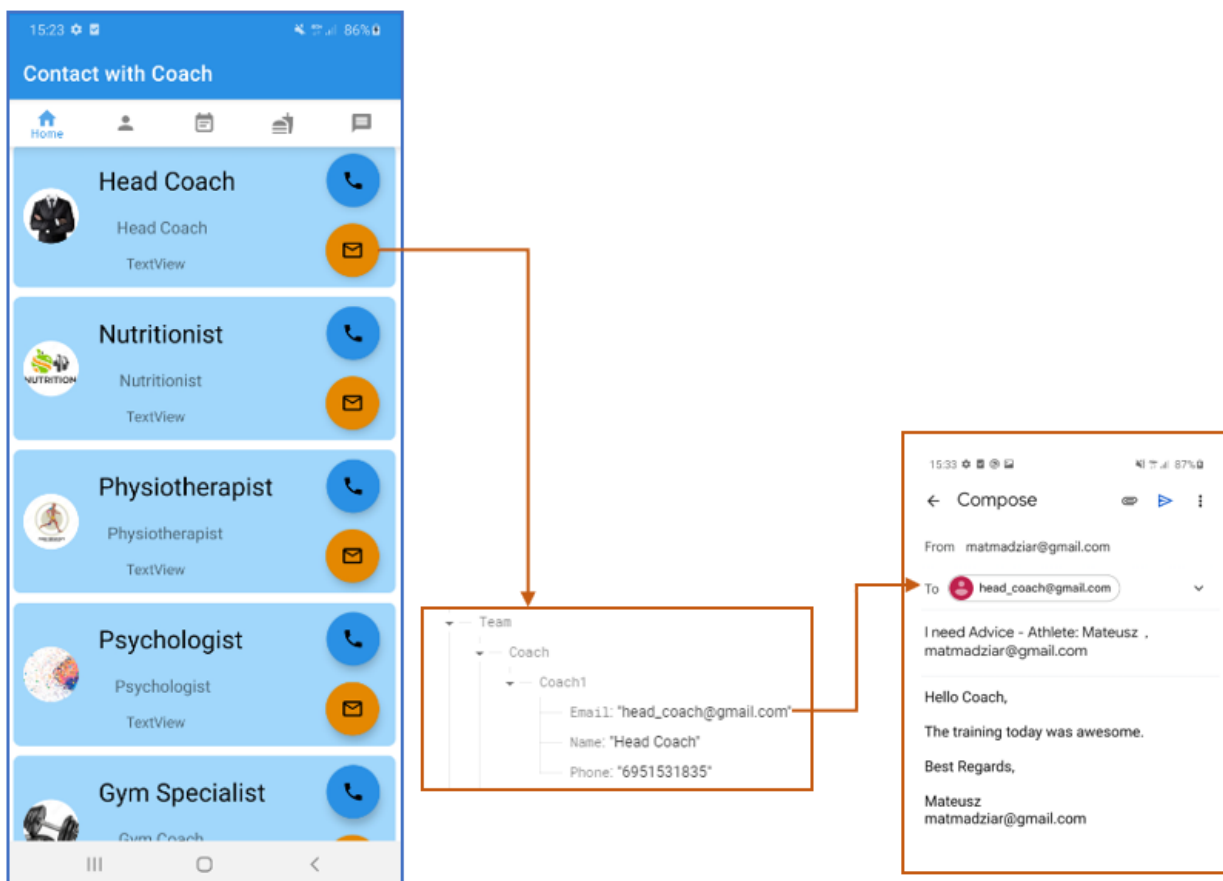
Εικόνα 3.77 Ομάδα υποστήριξης I



Εικόνα 3.78 Ομάδα υποστήριξης II

3.4.11.2 **Δυνατότητα αποστολής μηνύματος με χρήση ηλεκτρονικού ταχυδρομείου**

Με τη χρήση αυτής της επιλογής ο χρήστης έχει τη δυνατότητα άμεσης αποστολής μηνύματος μέσω ηλεκτρονικού ταχυδρομείου, όπου τα στοιχεία του προπονητή (ή οποιουδήποτε άλλου μέλους της ομάδας υποστήριξης), αλλά και του ίδιου του αθλητή συμπληρώνονται αυτόματα (Εικόνα 3.79).



Εικόνα 3.79 Αποστολή μηνύματος με τη χρήση ηλεκτρονικού ταχυδρομείου

Η συνάρτηση της εικόνας 3.80 εκτελείται κάθε φορά που ο χρήστης επιλέξει την αποστολή μηνύματος με τη χρήση ηλεκτρονικού ταχυδρομείου.

```

//Function For sending Emails
fun SendEmail(userAddress : String,userSubject : String, userMessage : String){

    val emailAddresses = arrayOf(userAddress)
    val emailIntent = Intent(Intent.ACTION_SEND)
    emailIntent.type = "*/*" // "*/*" is valid for all type of emails
    emailIntent.putExtra(Intent.EXTRA_EMAIL, emailAddresses)
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, userSubject)
    emailIntent.putExtra(Intent.EXTRA_TEXT, userMessage)

    if(emailIntent.resolveActivity(packageManager) != null){
        // in this intent we allow user to choose the alternatives to send the email
        // via email provider they wants (gmail,hotmail application)
        startActivity(Intent.createChooser(emailIntent, title: "Choose an app"))
    }
}
}

```

Εικόνα 3.80 Συνάρτηση SendEmail()

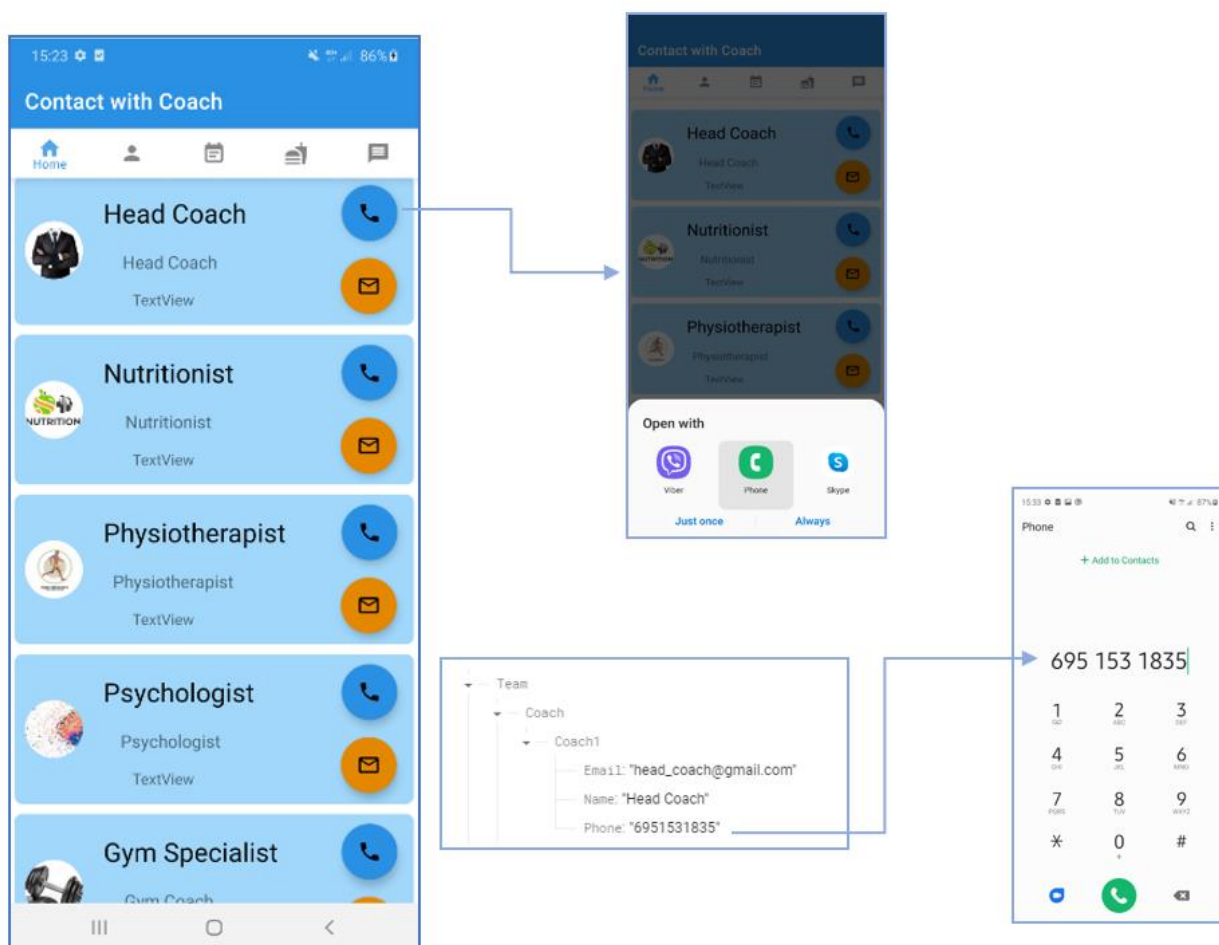
Για να μπορέσει να πραγματοποιηθεί η αποστολή μηνύματος είναι απαραίτητη η ρύθμιση της δραστηριότητας «ContactCoachActivity» στο Manifest File (εικόνα 3.81).

```
<activity
    android:name=".ContactCoachActivity"
    android:exported="false"
    android:theme="@style/SignupTheme">
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <data android:mimeType="*/*" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

Εικόνα 3.81 Απαραίτητη ρύθμιση στο Manifest File

3.4.11.3 Δυνατότητα πραγματοποίησης κλήσης

Είναι απαραίτητη η συγκατάθεση και η παραχώρηση δικαιωμάτων στην εφαρμογή να πραγματοποιεί κλήσεις. Αυτό γίνεται με την εισαγωγή της βιβλιοθήκης «**android.permission.CALL_PHONE**» στον Manifest File της εφαρμογής.



Εικόνα 3.82 Δυνατότητα πραγματοποίησης κλήσης

Με την επιλογή της πραγματοποίησης κλήσης, καλείται η συνάρτηση «StartCall()» που παρουσιάζεται στην εικόνα 3.83.

Στα πιο καινούρια μοντέλα κινητών με ανανεωμένη έκδοση του λογισμικού Android, είναι απαραίτητη η ένταξη μια δεύτερης συνάρτησης που εξασφαλίζει την παροχή δικαιωμάτων (συνάρτηση «onRequestPermissionsResult»).

Ο απαραίτητος κώδικας για την παροχή της δυνατότητας κλήσης παρουσιάζεται στην εικόνα 3.83.

```
// function for making a call
fun StartCall(userNumber:String){
    if(ContextCompat.checkSelfPermission( context: this, Manifest.permission.CALL_PHONE)≠ PackageManager.PERMISSION_GRANTED){
        // I specified the permission to be obtained from the user after it is not equal
        //This line is going to be visible in the first time (when the permission has never be given before)
        ActivityCompat.requestPermissions( activity: this, arrayOf(Manifest.permission.CALL_PHONE), requestCode: 100)
    }
    else{
        // this is to perform call. But is not going to work alone in newest versions of android.
        val phone : String = userNumber.toString()
        val intent = Intent(Intent.ACTION_VIEW) //
        intent.data = Uri.parse( uriString: "tel:$phone") // this shows the number when we call it
        startActivity(intent)
    }
}

// this is required for newest android version. after version 29 needed to add this function
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
    grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    // here we check if the permission is granted one more time
    if (requestCode ==100 && grantResults.size >0 && grantResults[0] == PackageManager.PERMISSION_GRANTED){
        val phone : String = userNumber.toString()// if I don't write this, random number will be visible on the call screen
        val intent = Intent(Intent.ACTION_VIEW)
        intent.data = Uri.parse( uriString: "tel:$phone") // is important to make a call
        startActivity(intent)
    }
}
```

Εικόνα 3.83 Συνάρτηση StartCall() & onRequestPermissionsResult()

3.4.12 Υπολογισμός θερμιδικών απαιτήσεων

3.4.12.1 Περιγραφή λειτουργίας

Οι απαιτούμενες ημερήσιες θερμίδες υπολογίζονται με βάση τον τύπο του **Harris-Benedict** που περιγράφεται στην ενότητα 1.2.2. Ο αθλητής εισάγει τα σωματομετρικά χαρακτηριστικά, επιλέγει τον βαθμό δραστηριότητας της συγκεκριμένης ημέρας και γίνεται μια προσέγγιση για τις ενεργειακές απαιτήσεις του (εικόνα 3.88).

* Αξίζει να σημειωθεί, ότι αν και ο συγκεκριμένος τύπος υπολογισμού προσδιορίζει τις απαιτούμενες θερμιδικές απαιτήσεις με μεγάλη ακρίβεια, δεν παύει αυτό να είναι μόνο μια εκτίμηση. Για να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα, απαιτείται επικοινωνία με

πιστοποιημένο διατροφολόγο, διότι κάθε άνθρωπος είναι διαφορετικός και απαιτείται ειδική προσέγγιση.

3.4.12.2 Πρόσθετες επιλογές I – Επικοινωνίας με τον διατροφολόγο

Μέσω αυτής της δραστηριότητας ο χρήστης έχει τη δυνατότητα άμεσης επικοινωνίας με το διατροφολόγο της ομάδας υποστήριξης (εικόνες 3.84, 3.85).

Για να μπορεί ο διατροφολόγος να έχει μια ολοκληρωμένη εικόνα της κατάστασης, αλλά και του στόχου του αθλούμενου, είναι απαραίτητη η συμπλήρωση της φόρμας που εμφανίζεται. Η φόρμα αποτελείται από 11 ερωτήσεις. Στοχεύοντας στη μέγιστη απόδοση, είναι εξίσου σημαντικό να απαντηθούν όλες οι ερωτήσεις και να γίνει αυτό με απόλυτη ειλικρίνεια.

11:04 86%

Contact with Nutritionist

HEALTHY LIVING

NUTRITION WORKOUT MINDFULNESS VITAMINS SPORTS BALANCE

Σωματομετρικά στοιχεία (βάρος, ύψος, ηλικία):
Write the necessary informations

Εξετάσεις αίματος (Γενική, φεριτίνη, Σίδηρος, Γλυκόζη, Γολληστερόλη ολική/LDD/HDL/VLDL, CPK):
Write the necessary informations

Ιστορικό υγείας (αν υπάρχει κάποια πάθηση όπως: αναιμία/ διαβήτης/ θυρεοειδής/ υψηλή πίεση κτλ.):
Write the necessary informations

Ωράριο εργασίας και είδος εργασίας (καθιστική ή συνεχής κίνηση):
Write the necessary informations

Ωράριο προπόνησης, χιλιμετρικός όγκος ανα εβδομάδα και εβδομαδιαίο πρόγραμμα:
Write the necessary informations

Κάποια δυσανεξία ή αποστροφή σε συγκεκριμένα τρόφιμα:
Write the necessary informations

Εικόνα 3.84 Επικοινωνία με διατροφολόγο I

Αγαπημένες τροφές:
Write the necessary informations

Συμπληρώματα διατροφής (ποιά και γιατί):
Write the necessary informations

Φάρμακα (ποιά και γιατί):
Write the necessary informations

Περιγραφή τωρινής ημερίσας διατροφής:
Write the necessary informations

Προσωπικός Στόχος:
Write the necessary informations

Nutritionist
Nutritionist
Live Healthy

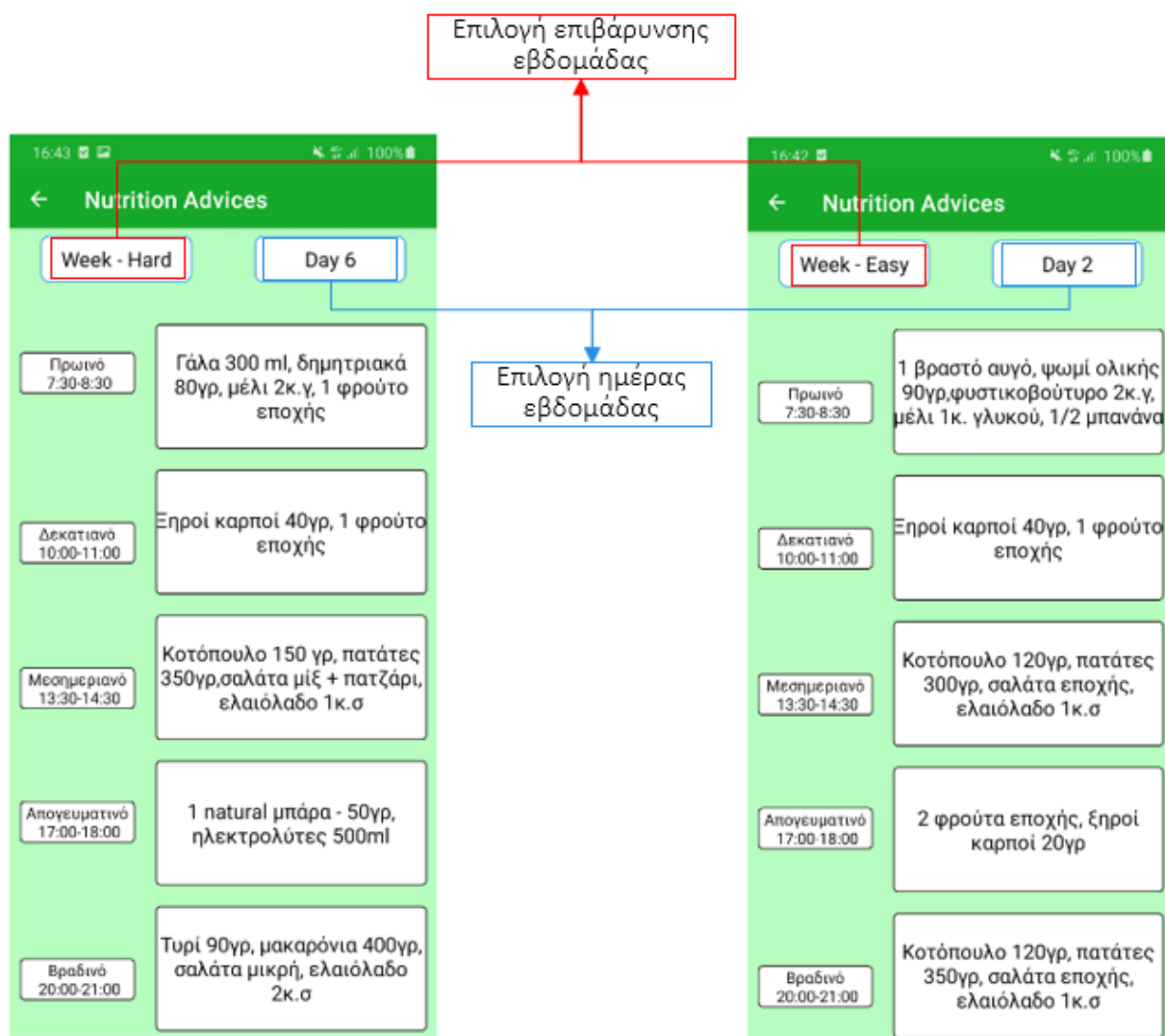
Call and Message icons

Εικόνα 3.85 Επικοινωνία με διατροφολόγο II

Επιπλέον, με σκοπό να διασφαλιστεί η ιδιωτικότητα του αθλητή, οι ερωτήσεις που αφορούν τις προσωπικές του πληροφορίες δεν καταγράφονται στη βάση δεδομένων. Με την επιλογή αποστολής μηνύματος, όλες οι απαντήσεις μεταφέρονται αυτόματα στην εφαρμογή ηλεκτρονικού ταχυδρομείου, μαζί με τα στοιχεία του χρήστη (που βρίσκονται στη βάση δεδομένων), όπου μπορεί να γίνει επανέλεγχος και αποστολή.

3.4.12.3 Πρόσθετες επιλογές II– Εβδομαδιαίο πρόγραμμα διατροφής

Σε αυτή τη δραστηριότητα παρέχεται ένα ενδεικτικό πρόγραμμα διατροφής για συγκεκριμένη μέρα, ανάλογα με την προπονητική επιβάρυνση (εικόνα 3.86). Παρέχονται επίσης σημαντικές συμβουλές διατροφής. Επιπλέον, παρέχονται ενδεικτικά εβδομαδιαία προγράμματα διατροφής, ανάλογα με την επιβάρυνση της εβδομάδας. Το πρόγραμμα διατροφής σχεδιάστηκε από ειδικό αθλητικό διατροφολόγο και είναι προσαρμοσμένο στο εβδομαδιαίο πρόγραμμα προπόνησης.



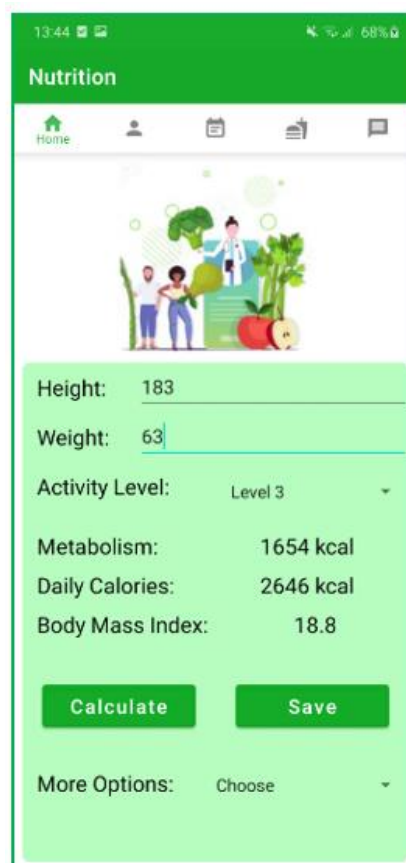
Εικόνα 3.86 Ενδεικτικό πρόγραμμα διατροφής

Ιδιαίτερα σημαντικό για την άμεση αποκατάσταση είναι λήψη γεύματος μετά την προπόνηση. Ο χρόνος μεταξύ της λήξης της προπόνησης και του πρώτου γεύματος μετά την προπόνηση δεν πρέπει να ξεπερνάει τα 30 λεπτά.

Στην Εικόνα 3.87 παρουσιάζονται διαφορετικές επιλογές ανάλογα με το τύπο της προπόνησης που εκτέλεσε ο αθλητής.



Εικόνα 3.87 Πληροφορίες για άμεση αποκατάσταση

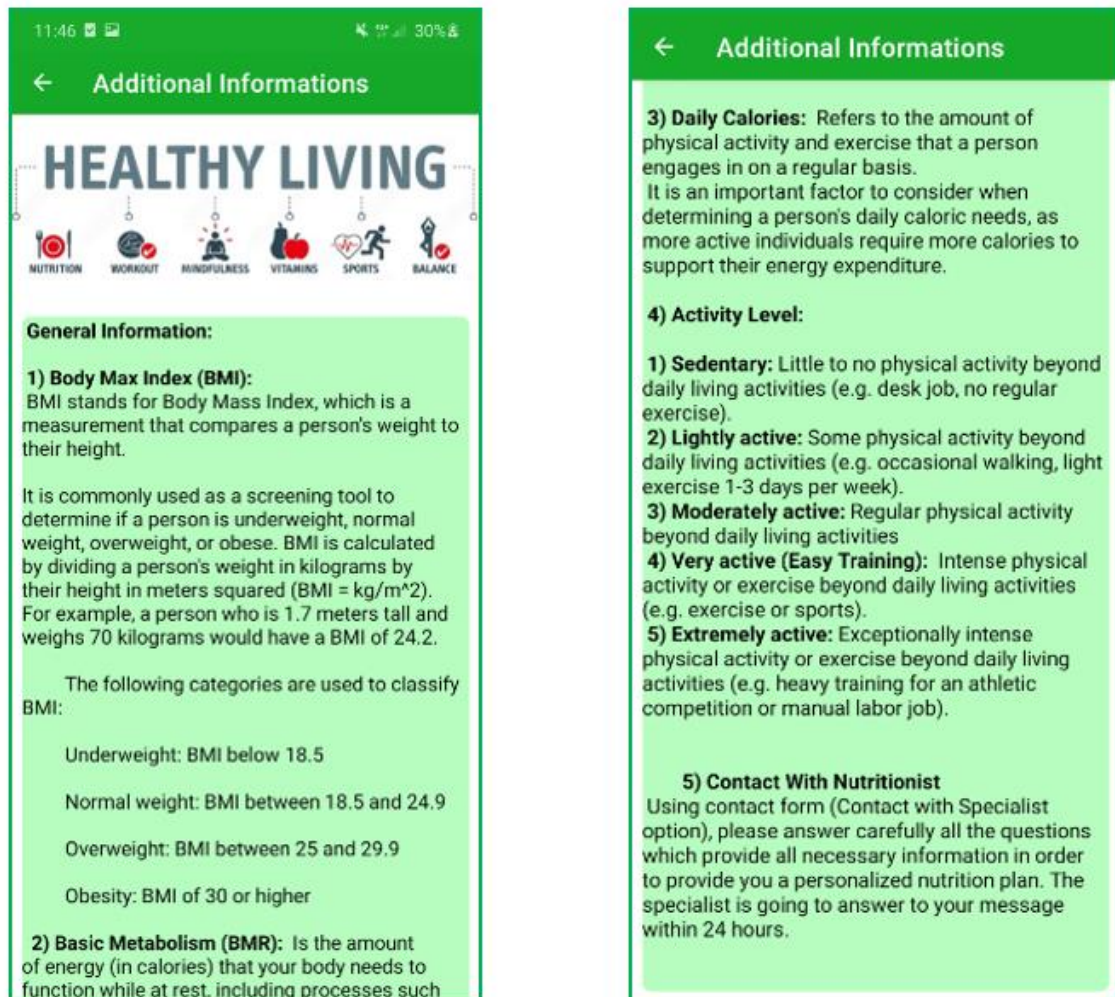


Εικόνα 3.88 Υπολογισμός θερμιδικών αναγκών

3.4.12.4 Πρόσθετες επιλογές III – Βασικές πληροφορίες διατροφής

Σε αυτή τη δραστηριότητα αναφέρονται οι βασικές έννοιες που σχετίζονται με την επιστήμη της Διατροφολογίας, με στόχο την ενημέρωση του αθλητή για τις έννοιες που εμφανίζονται στην εφαρμογή (Εικόνα 3.89).

Ο χρήστης ενημερώνεται για το πως να επιλέξει το επίπεδο της ημερήσιας δραστηριότητας (Activity Level) που επιτρέπει τον σωστό προσδιορισμό των απαιτούμενων θερμιδικών αναγκών (Εικόνα 3.88) της κάθε ημέρας.



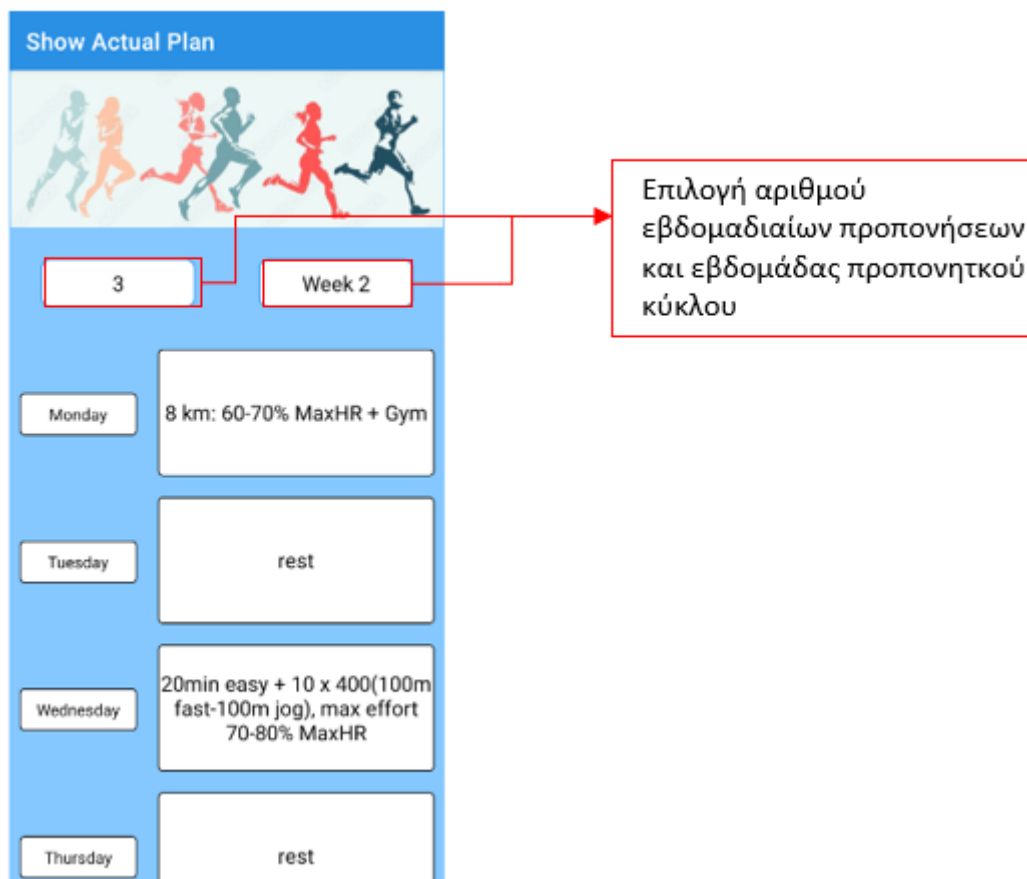
Εικόνα 3.89 Βασικές πληροφορίες διατροφής

3.4.13 Είσοδος προπονητή

3.4.13.1 Επισκόπηση προπονητικού προγράμματος

Μέσω αυτής της δραστηριότητας επιτρέπεται η επίβλεψη του είδη υπάρχοντος προπονητικού προγράμματος. Ο προπονητής επιλέγει τον αριθμό εβδομαδιαίων προπονήσεων και την εβδομάδα του προπονητικού κύκλου (εικόνα 3.90). Για την υλοποίηση χρησιμοποιήθηκαν 2 εξαρτημένοι spinners (dependent spinners).

* Το πρόγραμμα που εμφανίζεται και υπάρχει στη βάση δεδομένων σχεδιάστηκε από προπονητή με ειδικότητα στις μεσαίες και μεγάλες αποστάσεις ο οποίος είναι όχι μόνο πιστοποιημένος αλλά και πεπειραμένος. Σχεδιάστηκε με τέτοιο τρόπο ώστε να ικανοποιούνται όλες οι βασικές αρχές προπονητικής που αναφέρονται στο 1^ο κεφάλαιο.



Εικόνα 3.90 Εμφάνιση και τροποποίηση προπονητικού προγράμματος

3.4.13.2 *Επιλογή μορφοποίησης προπονητικού προγράμματος*

Όπως έχει αναλυθεί στην ενότητα 3.4.6, πρόσβαση σε αυτή την επιλογή έχει μόνο ο προπονητής, του οποίου τα στοιχεία βρίσκονται στη βάση δεδομένων. Σε διαφορετική περίπτωση ο χρήστης δεν έχει τη δυνατότητα να αλλάζει το προπονητικό πρόγραμμα ή να επιβλέπει την πορεία των υπόλοιπων αθλούμενων.

Διαφέρει με την επιλογή της ενότητας 3.4.13.1 διότι δίνει την επιλογή στο προπονητή να αλλάζει το προπονητικό πρόγραμμα. Όλοι οι αθλητές μπορούν να δουν την κάθε αλλαγή που έγινε σε πραγματικό χρόνο.

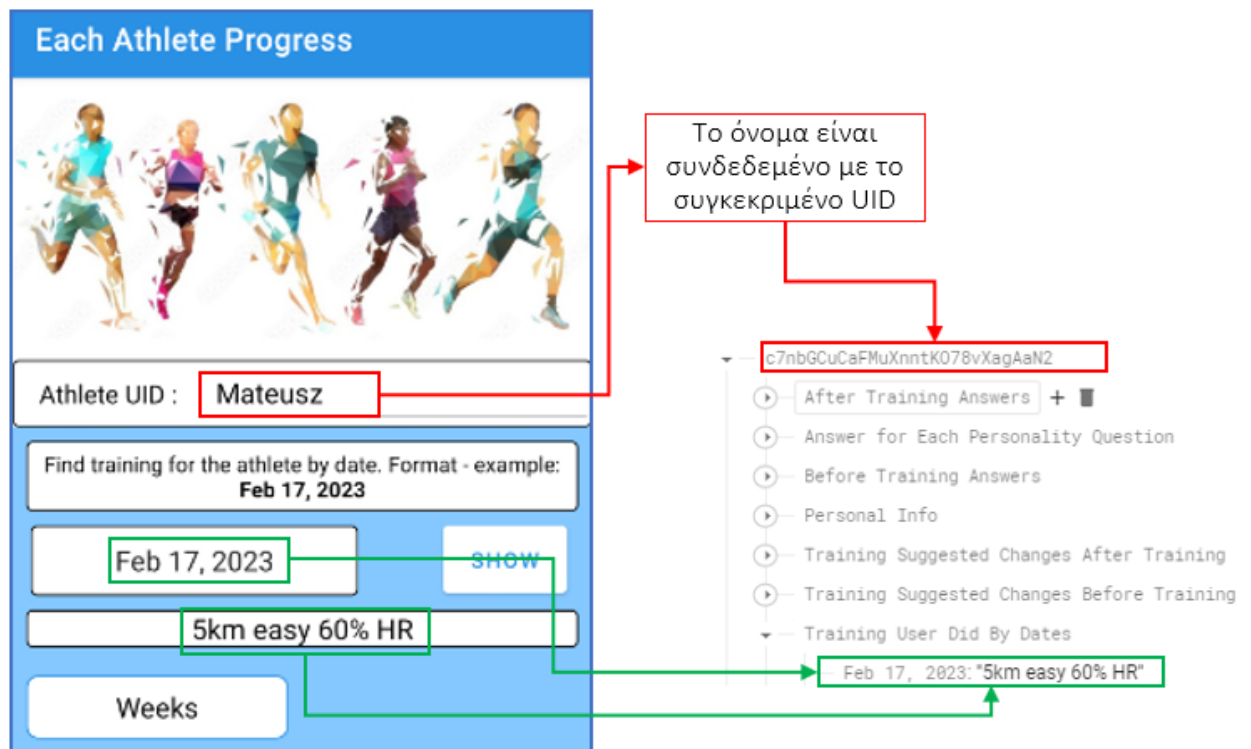
Η αλλαγή πραγματοποιείται με την επιλογή του αριθμού των εβδομαδιαίων προπονήσεων, αλλά και της εβδομάδας του προπονητικού κύκλου (Εικόνα 3.90). Για την υλοποίηση χρησιμοποιήθηκαν 2 εξαρτημένοι spinners (dependent spinners).

3.4.13.3 *Επίβλεψη δραστηριότητας επιλεγμένου αθλητή*

Με την εισαγωγή του ονόματος του αθλητή, έχει πρόσβαση σε όλες τις δραστηριότητες που εκτέλεσε ο αθλητής. Παρέχονται επίσης διαφορετικές επιλογές αναζήτησης (εικόνα 3.91).

Επιτρέπεται η αναζήτηση δεδομένων:

- με την εισαγωγή συγκεκριμένης ημερομηνίας,
- με την επιλογή ημέρας και εβδομάδας (για σύγκριση απόδοσης μεταξύ 2 ή περισσότερων προπονητικών κύκλων),
- αλλά και την απευθείας επισκόπηση προπονήσεων του τελευταίου κύκλου των 12 εβδομάδων, με επιλογή της εβδομάδας.

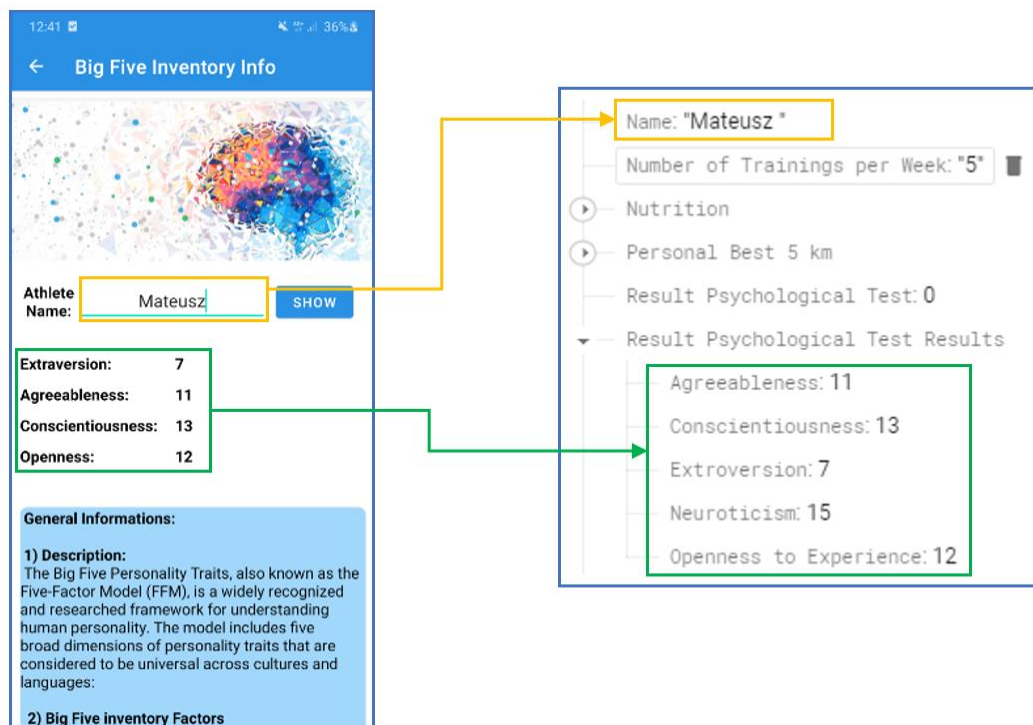


Εικόνα 3.91 Επίβλεψη δραστηριότητας συγκεκριμένου αθλητή

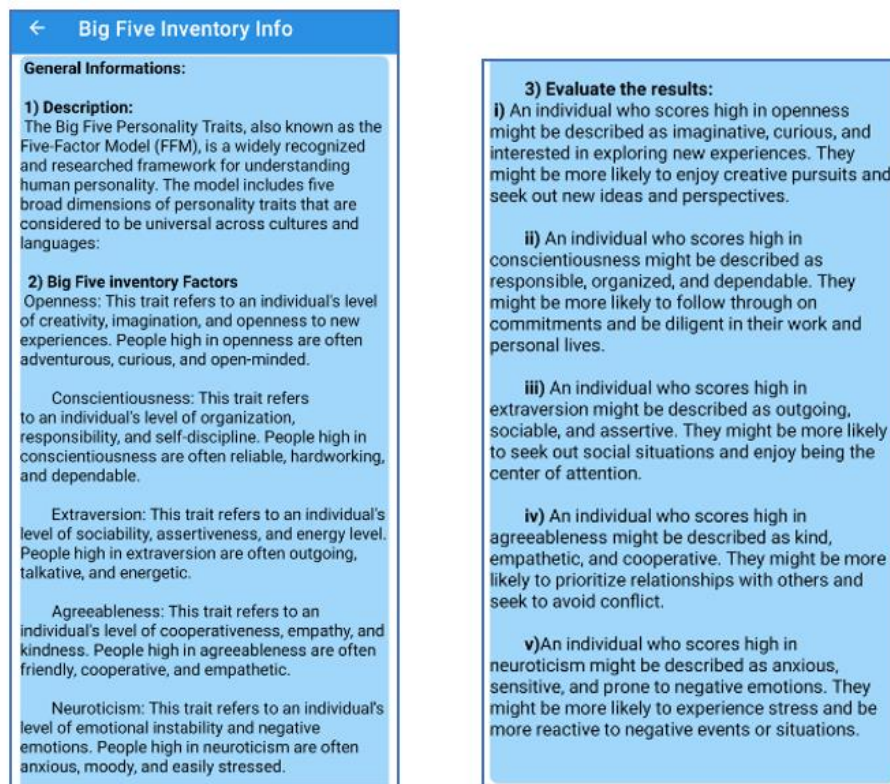
3.4.13.4 Πληροφορίες για τα αποτελέσματα του ψυχολογικού τεστ

Σε αυτή τη δραστηριότητα αναφέρονται πληροφορίες για το ψυχολογικό προφίλ του κάθε αθλητή. Με την εισαγωγή του ονόματος του αθλητή, εμφανίζονται τα στοιχεία για τα το κάθε επίπεδο προσωπικότητας του (Εικόνα 3.92).

Παρέχονται και οι απαιτούμενες πληροφορίες για τη σωστή ερμηνεία των αποτελεσμάτων του ψυχολογικού τεστ που συμπλήρωσε ο αθλητής (Εικόνα 3.93)



Εικόνα 3.92 Εμφάνιση αποτελεσμάτων ψυχολογικού τεστ



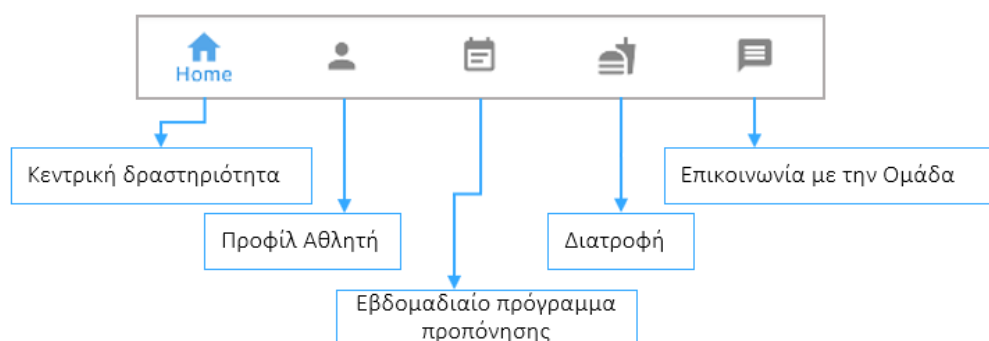
Εικόνα 3.93 Πληροφορίες για τα αποτελέσματα ψυχολογικού τεστ

3.4.14 Επιπλέον λειτουργίες – Navigation Menu

Για την καλύτερη πλοήγηση του χρήστη στην εφαρμογή, σημαντική είναι η δυνατότητα άμεσης πρόσβαση στις βασικές λειτουργίες της εφαρμογής. Ο κώδικας υλοποίησης παρουσιάζεται στις εικόνες 3.95 και 3.96.

- Κεντρικό δραστηριότητα (MainActivity)
- Προφίλ Αθλητή (AthleteProfileActivity)
- Εβδομαδιαίο πρόγραμμα προπόνησης (TrainingPlanActivity)
- Κεντρική δραστηριότητα διατροφής (NutritionActivity)
- Επικοινωνία με την ομάδα υποστήριξης (ContactWithTeamActivity)

Για αυτό το λόγο υλοποιήθηκε το Navigation Menu (εικόνα 3.94).



Εικόνα 3.94 Navigation Menu

```

navMenu = findViewById(R.id.bottomNavigationView)
navMenu.setOnItemSelectedListener { menuItem →
    when(menuItem.itemId){
        R.id.miMessage →{
            changeActivityToMessage()
            true *setOnItemSelectedListener
        }
        R.id.miProfile →{
            true *setOnItemSelectedListener
        }
        R.id.miHome →{
            changeActivityToHome()
            true *setOnItemSelectedListener
        }
        R.id.miPlan →{
            changeActivityToTrainingPlan()
            true *setOnItemSelectedListener
        }
        R.id.miNutrition →{
            changeActivityToNutrition()
            true *setOnItemSelectedListener
        }
        else → {
            false *setOnItemSelectedListener
        }
    }
}
    
```

Εικόνα 3.95 Navigation Menu - κώδικας I

```

private fun changeActivityToHome(){
    startActivity(Intent( packageContext: this,
        MainActivity :: class.java))
    finish()
}
private fun changeActivityToProfile(){
    startActivity(Intent( packageContext: this,
        AthleteProfileActivity :: class.java))
    finish()
}
private fun changeActivityToNutrition(){
    startActivity(Intent( packageContext: this,
        NutritionActivity :: class.java))
    finish()
}
private fun changeActivityToTrainingPlan(){
    startActivity(Intent( packageContext: this,
        TrainingPlanMainNewActivity :: class.java))
    finish()
}
private fun changeActivityToMessage(){
    startActivity(Intent( packageContext: this,
        ContactCoachActivity :: class.java))
    finish()
}
    
```

Εικόνα 3.96 Navigation Menu - κώδικας II

3.4.15 Ασφάλεια δεδομένων - Security Rules

Οι κανόνες ασφαλείας της Firebase συμβάλλουν στην προστασία ευαίσθητων δεδομένων, όπως προσωπικές πληροφορίες, από μη εξουσιοδοτημένη πρόσβαση. Στην εικόνα 3.97 περιγράφονται οι κανόνες ασφαλείας της βάσης δεδομένων που διασφαλίζουν την προστασία των δεδομένων της εφαρμογής που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής. Στο DB directory (Database Directory) “Athlete Info”, όπου υπάρχουν όλες οι προσωπικές πληροφορίες για κάθε αθλητή, για να μπορεί να γίνει πρόσβαση στις πληροφορίες αυτές, επιβάλλεται uid αυθεντικοποίηση να είναι ίδιο με το user ID που είναι το όνομα του υποφακέλου που περιέχει τα δεδομένα. Αυτό διασφαλίζει την εισαγωγή δεδομένων του κάθε χρήστη μόνο στον ειδικό φάκελο του. Αποτρέπει την εισαγωγή δεδομένων οποιουδήποτε άλλου αυθεντικοποιημένου χρήστη σε φάκελο με πληροφορίες διαφορετικού χρήστη. Επιτρέπεται όμως η αναγνώριση των δεδομένων από άλλους αυθεντικοποιημένους χρήστες. Μέσω αυτής της επιλογής επιτρέπεται η πρόσβαση του προπονητή στα βασικά στοιχεία του αθλητή. Η πλατφόρμα της Firebase παρέχει τη δυνατότητα προσομοίωσής τους σε κανονικές συνθήκες.

```

1 {
2   "rules":{
3     "Athlete Info":{
4       "$uid": {
5         "Personal Info":{
6           ".read": "auth.uid != null",
7           ".write": "auth.uid != null && $uid == auth.uid"
8         },
9         "After Training Answers":{
10          ".read": "auth.uid != null",
11          ".write": "auth.uid != null && $uid == auth.uid"
12        },
13        "Before Training Answers":{
14          ".read": "auth.uid != null",
15          ".write": "auth.uid != null && $uid == auth.uid"
16        },
17        "Training user Finally Did":{
18          ".read": "auth.uid != null",
19          ".write": "auth.uid != null && $uid == auth.uid"
20        },
21        "Training user Finally did Without Date":{
22          ".read": "auth.uid != null",
23          ".write": "auth.uid != null && $uid == auth.uid"
24        },
25        "Training User Did By Dates":{
26          ".read": "auth.uid != null",
27          ".write": "auth.uid != null && $uid == auth.uid"
28        },
29        "Answer for eachPersonality Question":{
30          ".read": "auth.uid != null",
31          ".write": "auth.uid != null && $uid == auth.uid"
32        }
33      }
34    },
35    "Before Training Questions":{
36      ".read": "auth.uid != null",
37      ".write": "auth.uid != null"
38    },
39    "After Training Questions":{
40      ".read": "auth.uid != null",
41      ".write": "auth.uid != null"
42    },
43    "Personality Questions":{
44      ".read": "auth.uid != null",
45      ".write": false
46    },
47    "Team":{
48      ".read": "auth.uid != null",
49      ".write": false
50    },
51    "Training Plan":{
52      ".read": "auth.uid != null",
53      ".write": "auth.uid != null"
54    }
55  }
56 }

```

Εικόνα 3.97 Security Rules

4 ΚΕΦΑΛΑΙΟ 4^ο : Επίλογος

4.1 Εισαγωγή κεφαλαίου

Σε αυτό το κεφάλαιο παρουσιάζεται η σύνοψη της εργασίας, αναλύονται τα προβλήματα που παρουσιάστηκαν κατά την υλοποίηση της εφαρμογής που αναπτύχθηκε, όσο και ο τρόπος αντιμετώπισης τους. Τέλος, παρουσιάζονται οι παρατηρήσεις και τα συμπεράσματα που προέκυψαν, καθώς και μερικές προτάσεις για μελλοντική εξέλιξη και βελτίωση της εφαρμογής.

4.2 Σύνοψη Εργασίας

Η βασική ιδέα της δημιουργίας της συγκεκριμένης εφαρμογής ήταν η συγκέντρωση όλων των απαραίτητων πληροφοριών που απαιτούνται για την επίτευξη καλύτερης δυνατής καθοδήγησης ενός δρομέα. Η εφαρμογή που υλοποιήθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας περιελάμβανε τις ακόλουθες διαδοχικές φάσεις υλοποίησης:

- Διαμόρφωση και παράθεση πρότασης
- Έρευνα του θεωρητικού υποβάθρου και βιβλιογραφική αναζήτηση
- Εκμάθηση της απαιτούμενης γλώσσας προγραμματισμού αλλά και όλων των απαιτούμενων τεχνικών εργαλείων (Kotlin, Firebase, Android Studio)
- Κατάλληλος σχεδιασμό της εφαρμογής (δομή της εφαρμογής, δομή της βάσης δεδομένων)
- Υλοποίηση και κατασκευή της εφαρμογής

Οι διαδοχικές φάσεις υλοποίησης του έργου μαζί με την αντίστοιχη χρονοσήμανση τους απεικονίζονται στην εικόνα 4.1.



Εικόνα 4.1 Σύνοψη εργασίας

4.3 Προβλήματα - Επίλυση

Κατά την υλοποίηση της εφαρμογής ανά διαστήματα παρουσιάστηκαν κάποια προβλήματα. Κατά κύριο λόγο αφορούσαν τον σχεδιασμό, τον προγραμματισμό, αλλά και την επικοινωνία των επιμέρους συστημάτων.

Το πρώτο πρόβλημα εμφανίστηκε αμέσως μετά την εγκατάσταση του Android Studio. Το συγκεκριμένο πρόγραμμα παρέχει πάρα πολλές επιλογές (ιδιαίτερα οι τελευταίες εκδόσεις του), όμως έχει αρκετά υψηλές απαιτήσεις όσο αφορά τις προδιαγραφές για έναν απλό φορητό υπολογιστή. Για αυτό το λόγο ήταν απαραίτητη η διενέργεια αναβαθμίσεων του (αντικατάσταση του σκληρού δίσκου από HDD σε SSD, αλλά και η πρόσθεση μνήμης RAM).

Όσο αφορά όμως την υλοποίηση της εφαρμογής, αρχικά η βάση δεδομένων σχεδιάστηκε σε μια αρκετά απλή δομή. Με την πορεία της εργασίας φάνηκε αναγκαία η αλλαγή της δομής της, αλλά και η ενσωμάτωση παραπάνω δεδομένων. Για την επίλυση των περισσότερων τεχνικών προβλημάτων που αφορούσαν τη σύνδεση με τη βάση δεδομένων μεγάλη βοήθεια ήταν τα επίσημα documentation της Firebase, αλλά και του Android Studio.

Για την επίτευξη της καλύτερης δυνατής προσαρμογής στις ανάγκες του χρήστη, ήταν απαραίτητη η παροχή της δυνατότητας επιλογής προγραμμάτων προπόνησης αλλά και διατροφής. Η λύση που επιλέχθηκε ήταν η ενσωμάτωση «dependent spinners» σε ορισμένες δραστηριότητες της εφαρμογής. Αυτό επιτρέπει την ανάκτηση συγκεκριμένων πληροφοριών που βρίσκονται στη βάση δεδομένων της Firebase. Αυτό είχε ως αποτέλεσμα την εξέλιξη της βάσης δεδομένων και την αύξηση των δεδομένων κατά τη διάρκεια της συγγραφής της διπλωματικής. Παρόμοια, και η αρχική δομή της εφαρμογής ήταν αρκετά απλή σε σχέση με το τελικό της στάδιο (εικόνα 3.1), γεγονός που απαιτούσε διαρκής αλλαγές.

Άλλη σημαντική δυσκολία που παρουσιάστηκε, ήταν πως εκτός από τις απαιτούμενες τεχνικές γνώσεις και τις γνώσεις προγραμματισμού, η υλοποίηση της συγκεκριμένης εφαρμογής απαιτούσε αρκετές γνώσεις εργοφυσιολογίας, προπονητικής, διατροφολογίας αλλά και αθλητικής ψυχολογίας. Για να μπορέσει η εφαρμογή να υποστηρίζεται επιστημονικά και τα αποτελέσματα να είναι ορθά, ήταν απαραίτητη η στενή συνεργασία με πιστοποιημένο προπονητή, διατροφολόγο, αλλά και αθλητικό ψυχολόγο. Συχνά μετά από την συνάντηση με την ομάδα υποστήριξης, είχαν διαπιστωθεί λάθη και ήταν απαραίτητη η ένταξη αλλαγών στη δομή της εφαρμογής (ή στη βάση δεδομένων).

4.4 Παρατηρήσεις – Συμπεράσματα

Η παρούσα εφαρμογή επίβλεψης και καθοδήγησης αθλητών είναι μια εφαρμογή με ιδιαίτερα ψηλό βαθμό δυσκολίας, καθώς οι παράμετροι που πρέπει να ελεγχθούν και να επεξεργαστούν είναι πάρα πολύ. Ο κάθε άνθρωπος είναι διαφορετικός και απαιτείται ιδιαίτερη μεταχείριση. Η εφαρμογή που υλοποιήθηκε φαίνεται να είναι ένα ιδανικό εργαλείο για την παροχή στον προπονητή μιας ολοκληρωμένης εικόνας της κατάστασης του αθλητή. Αξίζει να σημειωθεί όμως ότι ο προσδιορισμός του δείκτη κόπωσης απαιτεί περαιτέρω μελέτη με σκοπό την αποφυγή πιθανών σφαλμάτων.

Η υλοποίηση της εφαρμογής ξεκίνησε με βασικές γνώσεις για τις εφαρμογές Android και τον τρόπο υλοποίησής τους, οι οποίες αυξανόντουσαν διαρκώς κατά την πορεία της εργασίας.

Αυτό είχε ως αποτέλεσμα τον εντοπισμό αρκετών σφαλμάτων και μικρών λαθών καθ' όλη τη διάρκεια της διαδικασίας σχεδίασης και υλοποίησης, που έπρεπε να διορθωθούν στην πορεία.

Επιπλέον, η εφαρμογή υλοποιήθηκε με τη χρήση αποκλειστικά δραστηριοτήτων (activities). Με την αντικατάσταση ορισμένων δραστηριοτήτων με «fragments», η εφαρμογή γίνεται πιο δυναμική και μπορεί να προφέρει ένα ακόμα πιο εύχρηστο και φιλικό για τον χρήστη περιβάλλον.

Η πιο λεπτομερή ανάλυση και σχεδιασμός της βάσης δεδομένων είναι εξαιρετικά σημαντικός για την αποφυγή αλλαγών και διορθώσεων, γεγονός το οποίο έχει άμεση σχέση με τον συνολικό χρόνο υλοποίησης της εφαρμογής. Επιπλέον, στη περίπτωση που η εφαρμογή θα χρησιμοποιείται από μεγάλο αριθμό χρηστών απαιτείται μια διαφορετική προσέγγιση όσο αφορά την επιλογή της βάσης δεδομένων. Για τη συγκεκριμένη εφαρμογή ιδανική φαντάζει η επιλογή της Cloud Firestore Database της Firebase.

Μετά από τις πρώτες πειραματικές δοκιμές της εφαρμογής σε πραγματικές συνθήκες, παρατηρήθηκε αρκετά θετική ανταπόκριση από τους προπονητές, αλλά και από τους ίδιους τους αθλητές που τη χρησιμοποίησαν, γεγονός που δίνει αισιοδοξία για τη μελλοντική πορεία της εφαρμογής.

4.5 Προτάσεις Μελλοντικής Εξέλιξης

Το αμέσως επόμενο βήμα στην εξέλιξη της συγκεκριμένης εφαρμογής, αποτελεί η σύνδεση στη συστημάτων ασύρματης μεταφοράς δεδομένων (π.χ. παλμογράφος, αισθητήρες μέτρησης ταχύτητας, επιταχύνσεων, και λοιπών φυσιομετρικών δεδομένων) που επιτρέπουν την επίβλεψη της κατάστασης του αθλούμενου σε πραγματικό χρόνο.

Επίσης, η ενσωμάτωση περισσότερων ερωτήσεων πριν και μετά την προπόνηση, συμπεριλαμβάνοντας ερωτήσεις για την περιεκτικότητα γαλακτικού οξέος στο αίμα, θα βοηθήσουν στον καλύτερο προσδιορισμό του δείκτη κοπώσεως.

Επιπλέον, η εφαρμογή που υλοποιήθηκε στο πλαίσιο αυτής της διπλωματικής μπορεί να χρησιμοποιηθεί από τα περισσότερα μοντέλα που χρησιμοποιούν λογισμικό Android. Δυστυχώς, δεν μπορεί να χρησιμοποιηθεί σε κινητά που χρησιμοποιούν διαφορετικό λογισμικό. Η υλοποίηση της συγκεκριμένης εφαρμογής και για κινητά με διαφορετικό λογισμικό (όπως π.χ. IOS), είναι μια ακόμα πρόταση μελλοντικής εξέλιξης.

Επίσης η συνεχής τεχνολογική πρόοδος στην έρευνα που επιτελείται αναφορικά με τα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks –WSNs) [45-48] δίνει ακόμα μεγαλύτερες προοπτικές εξέλιξης στο αντικείμενο αυτής της εργασίας. Ειδικότερα, τα Ασύρματα Δίκτυα Σώματος (Wireless Body Area Networks -WBANs) μπορούν να παρέχουν συνεχή παροχή φυσιομετρικών στοιχείων χωρίς μάλιστα να απαιτείται η ύπαρξη μπαταρίας για τους αισθητήρες [49].

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζει το ενδεχόμενο ενσωμάτωσης νευρωνικού δικτύου, το οποίο να προσαρμόζει το πρόγραμμα προπόνησης, με πλήρως αυτοματοποιημένο τρόπο, αλλά και με μεγαλύτερη ακρίβεια για τον κάθε αθλητή. Η ταυτόχρονη μάλιστα αξιοποίηση και φυσιομετρικών δεδομένων θα επαυξήσει την αποδοτικότητα του όλου συστήματος [50]. Αν αυτός ο στόχος επιτευχθεί ο ρόλος του προπονητή θα περιοριστεί σημαντικά.

Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

- [1] Kotlin
<https://developer.android.com/kotlin/>
- [2] Android Studio
<https://developer.android.com/studio>
- [3] Firebase Authentication
<https://developers.google.com/android/guides/client-auth>
- [4] Authenticate User with Google Account
<https://firebase.google.com/docs/auth/android/google-signin>
- [5] Manage User in Firebase
<https://firebase.google.com/docs/auth/android/manage-users>
- [6] Connect Android Studio with Firebase
<https://firebase.google.com/docs/database/android/start>
- [7] Structure Data
<https://firebase.google.com/docs/database/android/structure-data>
- [8] Read and Write data on Android
<https://firebase.google.com/docs/database/android/read-and-write>
- [9] Structure Security Rules
<https://firebase.google.com/docs/database/security/core-syntax>
- [10] Use conditions in Realtime Database Security Rules
<https://firebase.google.com/docs/database/security/rules-conditions>
- [11] Read and Write Data on Android
<https://firebase.google.com/docs/database/android/read-and-write>
- [12] Database Options in Firebase Database
<https://firebase.google.com/docs/database/rtdb-vs-firestore>
- [13] Git Commands
<https://dzone.com/articles/top-20-git-commands-with-examples>
- [14] Images for the project
<https://stock.adobe.com/>
- [15] Android App Development with Kotlin | Beginner to Advanced Course
<https://www.udemy.com/course/android-app-development-with-kotlin-beginner-to-advanced/>
- [16] The Complete Android Kotlin Developer Course | Kotlin A-Z™
<https://www.udemy.com/course/the-complete-android-kotlin-developer-course-kotlin-a-ztm/>
- [17] Migrate from Kotlin synthetics to Jetpack view binding
<https://developer.android.com/topic/libraries/view-binding/migration#groovy>
- [18] Kotlin Basics
<https://www.digitalocean.com/community/tutorials/kotlin-let-run-also-apply-with>
- [19] Color Palettes
<https://www.canva.com/colors/color-palettes/>
- [20] Big Five Inventory
<https://fetzer.org/sites/default/files/images/stories/pdf/selfmeasures/Personality-BigFiveInventory.pdf>

- [21] John, O. P., & Srivastava, S. (1999). The Big-Five trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin & O. P. John (Eds.), Handbook of personality: Theory and research (Vol. 2, pp. 102–138). New York: Guilford Press.
<https://pages.uoregon.edu/sanjay/pubs/bigfive.pdf>
- [22] Measuring the Big Five Personality Domains
<https://pages.uoregon.edu/sanjay/bigfive.html>
- [23] Different Type of Psychological Test
<https://openpsychometrics.org/>
- [24] Małgorzata Fortuna, „Podstawy Kształtowania i kontroli Zdolności Wysiłkowej Tlenowej i Beztlenowej”, Kolegium Karkonoskie w Jeleniej Górze, Jelenia Góra 2008
https://dbc.wroc.pl/Content/2550/PDF/Fortuna_podstawy_kształtowania.pdf
- [25] Βασιλῆς Κλεισούρας, „Εργοφυσιολογία” – Δέκατη Έκδοση Αναδομημένη Ι – Ιατρικές εκδόσεις Π.Χ. Πασχαλίδης - Καθηγητής Εργοφυσιολογίας Πανεπιστημίων Αθηνών & McGill, 2004
- [26] Górski J., „Fizjologiczne podstawy wysiłku fizycznego”, 2002, PZWL Warszawa 28. Helgerud J., VO₂ , próg beztlenowy i ekonomika biegu u maratończyków – mężczyzn i kobiet” SportWyczynowy, 11-12, 73-81
- [27] William D. Mc Ardle, Frank I.Katch, Victor L.Katch, “Exercise Physiology: Nutrition, Energy, and Human Performance”
- [28] Hill D., Smith J., „Gender difference in anaerobic capacity: role of aerobic contribution”, British Journal of Sports Medicine, 1, 45-48, 1993
- [29] Bunc V., Heller J., Zdanowicz R., „Sposoby wyznaczania progu przemian beztlenowych”, SportWyczynowy, 1983, 7-8, 14
- [30] Chmura J. „Środki treningowe w kształtowaniu wytrzymałości tlenowej”, Zakład Motoryczności Gracza, AWF Wrocław, wykład, 2007
- [31] Σταύρος Τζιωρτζής, „Αθλητικοί Δρόμοι”, Αθήνα 2004
- [32] Oliver Lafay, „Methode de Nutrition” “Skuteczne odżywianie w treningu i sporcie”, Oficyna Wydawnicza READ ME Włodzimierz Bińczyk sp.j
- [33] Firebase - Realtime Database
<https://firebase.google.com/products-build>
- [34] Firebase - Cloud Storage
<https://firebase.google.com/products-build>
- [35] Firebase – Setup
<https://firebase.google.com/docs/android/setup>
- [36] Firebase -Security Rules
<https://firebase.google.com/docs/rules/>
- [37] Firebase Authentication
<https://firebase.google.com/docs/auth>
- [38] Melvin H. Williams, “Nutrition for Health, Fitness & Sport, Fifth Edition”, Old Dominion University, Broken Hill publishers Ltd
- [39] The VO₂max calculator that is based on resting heart rate is derived from Uth N, Sorensen H, Overgaard K, Pedersen PK. Estimation of VO₂max from the ratio between HRmax and HRrest--the Heart Rate Ratio Method. Eur J Appl Physiol. 2004 Jan;91(1):111-5.
- [40] The three VO₂max calculators requiring physical exertion are adapted from Fahey, T., Insel, P., Roth, W., Fit & Well: Core Concepts and Labs in Physical Fitness and Wellness (Seventh Edition), McGraw-Hill, 2007.

- [41] The estimation of maximum heart rate for the resting heart rate based VO₂max calculator is derived from Tanaka, H., Monhan, K.D., Seals, D.G., Age-predicted maximal heart rate revisited. *Am Coll Cardiol* 2001; 37:153-156.
- [42] Εικόνα Catapult 1
http://bestperformancegroup.com/?page_id=1413
- [43] Εικόνα Catapult 2
<https://screenrant.com/catapult-one-wearable-vest-professional-performance-metrics-available/>
- [44] Εικόνα Garmin
<https://garmin.gr/p-50>
- [45] «Σχεδίαση, Ανάπτυξη και Προγραμματισμός, μέσω Τεχνητής Νοημοσύνης Έλεγχος Ρομποτικού Συστήματος για Εφαρμογές Ασφάλειας», Φωτεινή Κολίτση, Αθήνα-Αιγάλεω 2022
- [46] Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of Wireless Sensor Networks : An Up - to - Date Survey. **2020**, 1–24, doi:10.3390/asi3010014.
- [47] Nakas, C.; Kandris, D.; Visvardis, G. Energy Efficient Routing in Wireless Sensor Networks: A Comprehensive Survey. *Algorithms* **2020**, 13, doi:10.3390/a13030072.
- [48] Rountos, D., Kandris, D., Evangelakos, E. A., Tselikis, G. (2022, December). Energy Efficient Routing in Wireless Sensor Networks: A Comparative Study on LEACH Protocol and its Successors. In 2022 Panhellenic Conference on Electronics & Telecommunications (PACET) (pp. 1-6). IEEE.
- [49] Evangelakos, E.A.; Kandris, D.; Rountos, D.; Tselikis, G.; Anastasiadis, E. Energy Sustainability in Wireless Sensor Networks: An Analytical Survey. *J. Low Power Electron. Appl.* **2022**, 12, doi:10.3390/jlpea12040065.
- [50] Korovesis, N., Kandris, D., Koulouras, G., Alexandridis, A. (2019). Robot motion control via an EEG-based brain–computer interface by using neural networks and alpha brainwaves. *Electronics*, 8(12), 1387.