



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόγραμμα Μεταπτυχιακών Σπουδών
«ΠΛΗΡΟΦΟΡΙΚΗΣ»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Εφαρμογής Android – Διαιτολόγιο - FitDiet Development of Android Application – Diet Plan – FitDiet
Όνοματεπώνυμο Φοιτητή	Καμαδέλη Αγγελική
Πατρώνυμο	Θεόδωρος
Αριθμός Μητρώου	ΜΠΠΛ18023
Επιβλέπων	Ευθύμιος Αλέπης, Καθηγητής

Ημερομηνία Παράδοσης **Ιούνιος 2024**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

Πίνακας Περιεχομένων

Περίληψη	5
Abstract	5
Ευχαριστίες	6
1. Εισαγωγή	7
1.1. Σύντομη Περιγραφή Προβλήματος	7
1.2. Σκοπός και Στόχοι	7
2. Βιβλιογραφική ανασκόπηση	9
2.1. Υπάρχουσες Λύσεις	9
2.2. Επιστημονικές Βάσεις	13
2.2.1. Διατροφικές Ανάγκες και Θερμίδες.....	13
2.2.2. Υπολογισμός Ημερήσιας Κατανάλωσης Θερμίδων	13
2.2.3. Σημασία της Άσκησης	14
2.2.4. Εξατομικευμένες Προτάσεις Διατροφής	15
2.2.5. Συνεργασία Διαιτολόγων και Εφαρμογών.....	15
2.2.6. Τεχνολογίες Παρακολούθησης Υγείας.....	15
3. Μεθοδολογία	17
3.1. Ανάλυση Απαιτήσεων	17
3.1.1. Λειτουργικές Απαιτήσεις	17
3.1.2. Μη Λειτουργικές Απαιτήσεις	17
3.2. Σχεδιασμός Συστήματος	17
3.2.1. Αρχιτεκτονική Συστήματος	17
3.2.2. Τεχνολογίες και Εργαλεία	18
3.2.3. Δομή της Βάσης Δεδομένων	18
3.2.4. Άλλες Τεχνολογίες και Εργαλεία	18
3.3. Περιγραφή Λειτουργιών	18
4. Ανάπτυξη Εφαρμογής	20
4.1. Περιγραφή Ανάπτυξης	20
4.2. Τεχνολογίες και Εργαλεία	20
5. Ανάλυση της Εφαρμογής	22
5.1. Εγκατάσταση και Ρυθμίσεις:	22
5.2. Παρουσίαση και τεχνική ανάλυση της Εφαρμογής Fitdiet	23
5.2.1. Οθόνη εγγραφής και σύνδεσης χρήστη - FacebookActivity	24
5.2.2. Interface επικοινωνίας των Fragments - FragmentsCommunicator	27
5.2.3. Κύρια Οθόνη - MainActivity	27
5.2.4. Επιλογή Φύλου χρήστη - Fragment1	32
5.2.5. Οθόνη Επιλογής επιπέδου φυσικής δραστηριότητας χρήστη - Fragment233	
5.2.6. Οθόνη Επιλογής στόχου χρήστη- Fragment6.....	35
5.2.7. Οθόνη για Καταχώρηση ηλικίας, ύψους και βάρους χρήστη - Fragment337	
5.2.8. Οθόνη Υπολογισμός και εμφάνιση ΔΜΣ, Ημερήσια κατανάλωση νερού και θερμίδων - Fragment4	40

5.2.9. FirebaseManager	43
5.2.10.Οθόνη Υπενθύμισης κατανάλωσης νερού - RemindersActivity.....	44
5.2.11.Οθόνη υπολογισμού θερμίδων - ApiActivity	47
5.2.12.Οθόνη Συνταγών -Recipes.....	50
5.2.13.Recipe	53
5.2.14.Οθόνη για Λεπτομέρειες συνταγής - RecipeDetailsFragment	54
5.2.15.Οθόνη για Πλάνο διατροφής - MealPlanFragment	56
5.2.16.Οθόνη για Λίστα Κατηγοριών Ασκήσεων - LearnActivity	58
5.2.17.YoutubeAPI.....	59
5.2.18.Οθόνη με Λίστα για συγκεκριμένη κατηγορία ασκήσεων - Exercise.....	60
5.2.19.Οθόνη -Tutorial άσκησης από το YouTube - TutorialPlayback.....	61
5.2.20.Οθόνη Σχετικά με εμάς - AboutFragment	63
6. Συμπεράσματα και Μελλοντική Εργασία	66
6.1. Συμπεράσματα:	66
6.2. Περιορισμοί και Προτάσεις Βελτίωσης.....	66
Βιβλιογραφία	68

Περίληψη

Η συνεχής εξέλιξη της τεχνολογίας και η διάδοση των κινητών τηλεφώνων με πληθώρα εφαρμογών έχουν μεταμορφώσει την καθημερινότητά μας, παρέχοντας εργαλεία που βελτιώνουν και διευκολύνουν την οργάνωση της ζωής μας. Παρά την πρόοδο αυτή, η διατήρηση της υγείας και της φυσικής κατάστασης απαιτεί στοχευμένες και πρακτικές λύσεις που μπορούν να ενσωματωθούν εύκολα στην καθημερινή ρουτίνα.

Η παρούσα εργασία έχει ως στόχο να προάγει τη σωστή διατροφή και τη σωματική δραστηριότητα μέσω μιας καινοτόμου εφαρμογής για κινητά. Η εφαρμογή επιτρέπει στους χρήστες να δημιουργούν εξατομικευμένα προγράμματα άσκησης και διατροφής, παρέχοντας πληροφορίες για τα γεύματα και το διατροφικό τους περιεχόμενο, καθώς επίσης και να παρακολουθούν εκπαιδευτικά βίντεο γυμναστικής. Με τον τρόπο αυτό, οι χρήστες μπορούν να διατηρούν μια ισορροπημένη διατροφή και να βελτιώνουν τη φυσική τους κατάσταση.

Επιπλέον, η εφαρμογή χρησιμοποιεί σύγχρονες τεχνολογίες και πλατφόρμες ανάπτυξης για να εξασφαλίσει μια φιλική προς τον χρήστη εμπειρία και παρέχει συνεχή παρακολούθηση για βέλτιστα αποτελέσματα. Τέλος, η εργασία εξετάζει τα συμπεράσματα και τις μελλοντικές προοπτικές της εφαρμογής, με σκοπό να καταστεί προσβάσιμη και χρήσιμη για όλους τους ανθρώπους, ανεξαρτήτως ηλικίας, σωματότυπου και προτιμήσεων άσκησης.

Abstract

The continuous advancement of technology and the widespread use of mobile phones with numerous applications have transformed our daily lives, providing tools that enhance productivity and facilitate organization. Despite this progress, maintaining health and physical fitness requires targeted and practical solutions that can be easily integrated into everyday routines.

This project aims to promote proper nutrition and physical activity through an innovative mobile application. The application allows users to access educational fitness videos and create personalized workout and meal plans, providing information on meals and their nutritional content. This enables users to maintain a balanced diet and improve their physical fitness.

Additionally, the application leverages modern technologies and development platforms to ensure a user-friendly experience and provides continuous monitoring for optimal results. Finally, the project examines conclusions and prospects of the application, with the aim of making it accessible and useful to all individuals, regardless of age, body type, and exercise preferences.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους όσους συνέβαλαν στην ολοκλήρωση αυτής της εργασίας,

Πρώτα απ' όλα, ευχαριστώ την οικογένειά μου για την αδιάκοπη στήριξη και ενθάρρυνση καθ' όλη τη διάρκεια των σπουδών μου. Η υπομονή και η πίστη τους στις ικανότητές μου υπήρξαν καθοριστικές για την επιτυχία μου. Ευχαριστώ επίσης τους συμφοιτητές μου για την πολύτιμη συντροφιά και υποστήριξη. Οι συζητήσεις και οι συνεργασίες μας ήταν πηγή έμπνευσης και μάθησης.

Ιδιαίτερες ευχαριστίες οφείλω στους καθηγητές μου, οι οποίοι με καθοδήγησαν με τις γνώσεις και την εμπειρία τους.

Τέλος, θα ήθελα ευχαριστήσω και τον επιβλέποντα καθηγητή μου, κύριο Ευθύμιο Αλέπη για την καθοδήγηση και τις συμβουλές του τόσο στις εργασίες των μαθημάτων, όσο και στη σύνταξη και βελτίωση αυτής της διπλωματικής. Η υποστήριξή του ήταν ανεκτίμητη για την εξέλιξη και ολοκλήρωση αυτής της εργασίας.

1. Εισαγωγή

1.1. Σύντομη Περιγραφή Προβλήματος

Η υγιεινή διατροφή είναι θεμελιώδης για τη διατήρηση της καλής υγείας και της ευημερίας. Περιλαμβάνει την κατανάλωση μιας ποικιλίας τροφίμων, τον έλεγχο της πρόσληψης λιπαρών, ζάχαρης και αλατιού, καθώς και την εξασφάλιση επαρκούς πρόσληψης βιταμινών και ανόργανων στοιχείων. Η σωστή διατροφή βοηθά στη διατήρηση ενός υγιούς σωματικού βάρους και μειώνει τον κίνδυνο εμφάνισης χρόνιων ασθενειών.¹

Η διατήρηση ενός υγιεινού τρόπου ζωής, στην εποχή μας είναι μια πρόκληση που αντιμετωπίζουμε όλοι. Οι γρήγοροι ρυθμοί της καθημερινότητας, η έλλειψη χρόνου και οι πολλαπλές υποχρεώσεις συχνά καθιστούν δύσκολη την παρακολούθηση της διατροφής και της σωματικής άσκησης. Επιπλέον, η πληθώρα πληροφοριών σχετικά με τη διατροφή και την υγεία μπορεί να προκαλέσει σύγχυση, δυσκολεύοντας την επιλογή των σωστών αποφάσεων².

1.2. Σκοπός και Στόχοι

Ο στόχος αυτής της διπλωματικής εργασίας είναι η δημιουργία μιας καινοτόμου εφαρμογής για κινητές συσκευές, η οποία θα λειτουργεί ως προσωπικός διαιτολόγος. Μέσω της εφαρμογής, οι χρήστες θα μπορούν να εισάγουν τα προσωπικά τους στοιχεία όπως ύψος, ηλικία, βάρος και επίπεδο φυσικής δραστηριότητας, για να υπολογίζουν τις ημερήσιες θερμιδικές τους ανάγκες³ (Nestle, 2013). Επιπλέον, θα παρέχεται η δυνατότητα καταγραφής της κατανάλωσης τροφίμων και θα προτείνονται εξατομικευμένες συνταγές και προγράμματα άσκησης⁴ (Dietary Guidelines Advisory Committee, 2015).

Οι στόχοι της εφαρμογής περιλαμβάνουν:

1. **Υπολογισμός Ημερήσιων Θερμιδικών Αναγκών:** Βάσει των προσωπικών δεδομένων του χρήστη, η εφαρμογή θα υπολογίζει τις απαιτούμενες θερμίδες για την επίτευξη ή διατήρηση του βάρους⁵ (Mifflin et al., 1990).
2. **Καταγραφή Κατανάλωσης Τροφίμων:** Οι χρήστες θα μπορούν να καταγράφουν τα τρόφιμα που καταναλώνουν καθημερινά και να παρακολουθούν τις θερμίδες που λαμβάνουν⁶ (Thompson & Subar, 2013).
3. **Προτάσεις Άσκησης:** Η εφαρμογή θα προτείνει εξατομικευμένα προγράμματα άσκησης ανάλογα με την κατηγορία εκγύμνασης που επιθυμεί ο χρήστης για την

¹ World Health Organization. (2020). "Healthy diet." Retrieved from <https://www.who.int/news-room/fact-sheets/detail/healthy-diet>.

² Swinburn, B., et al. (2011). "The global obesity pandemic: shaped by global drivers and local environments." *The Lancet*, 378(9793), 804-814

³ Nestle, M. (2013). *Food Politics: How the Food Industry Influences Nutrition and Health*. University of California Press.

⁴ Dietary Guidelines Advisory Committee. (2015). "Scientific Report of the 2015 Dietary Guidelines Advisory Committee." Office of Disease Prevention and Health Promotion.

⁵ Mifflin, M. D., et al. (1990). "A new predictive equation for resting energy expenditure in healthy individuals." *The American Journal of Clinical Nutrition*, 51(2), 241-247.

⁶ Thompson, F. E., & Subar, A. F. (2013). "Dietary assessment methodology." *Nutrition in the Prevention and Treatment of Disease*, 5(1), 3-39.

επίτευξη των θερμιδικών στόχων και θα προβάλει tutorial για την επίτευξη της άσκησης αυτής⁷.

4. **Εξατομικευμένες Προτάσεις Διατροφής:** Θα παρέχονται συνταγές και διατροφικά πλάνα προσαρμοσμένα στις ανάγκες και προτιμήσεις του χρήστη⁸ (Tessier et al., 2008).

Αυτή η εφαρμογή στοχεύει να προσφέρει ένα πρακτικό εργαλείο στους χρήστες για την επίτευξη και διατήρηση ενός υγιεινού τρόπου ζωής, συνδυάζοντας επιστημονικά δεδομένα με την καθημερινή πρακτική⁹ (U.S. Department of Health and Human Services, 2020).

⁷ American College of Sports Medicine. (2014). ACSM's Guidelines for Exercise Testing and Prescription. Lippincott Williams & Wilkins.

⁸ Tessier, S., et al. (2008). "Analysis of dietary data from food diaries." *Epidemiologic Methods*, 1(1), 107-113.

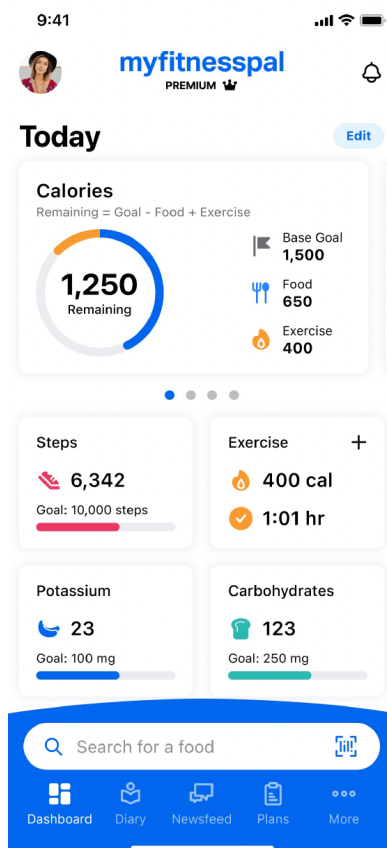
⁹ U.S. Department of Health and Human Services. (2020). "Dietary Guidelines for Americans 2020-2025." Retrieved from <https://www.dietaryguidelines.gov>.

2. Βιβλιογραφική ανασκόπηση

2.1. Υπάρχουσες Λύσεις

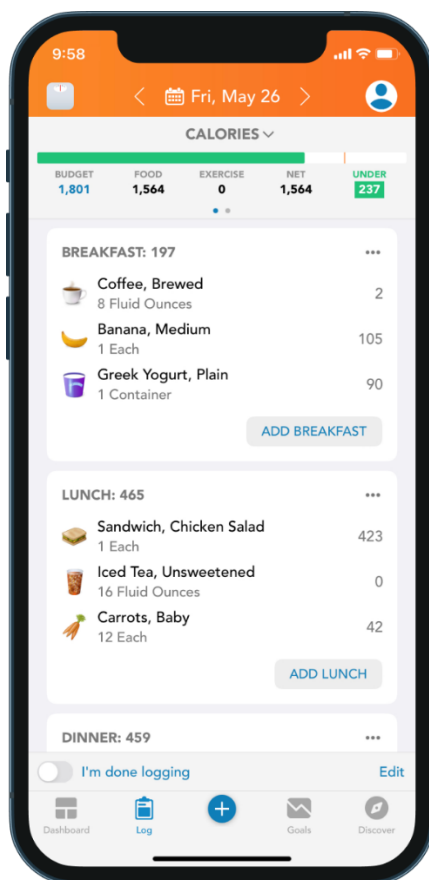
Στην τρέχουσα αγορά, υπάρχουν αρκετές εφαρμογές και πλατφόρμες που έχουν αναπτυχθεί για τη διαχείριση της διατροφής και της άσκησης. Μερικές από τις πιο δημοφιλείς εφαρμογές περιλαμβάνουν το MyFitnessPal, το Lose It!, το Noom, το FatSecret, το MyPlate Calorie Tracker & το Fitbit.

- **MyFitnessPal:** Αυτή η εφαρμογή επιτρέπει στους χρήστες να καταγράφουν την πρόσληψη τροφίμων και τις θερμίδες τους, παρέχοντας μια εκτεταμένη βάση δεδομένων τροφίμων και ένα εύκολο στη χρήση interface. Επιπλέον, προσφέρει δυνατότητες καταγραφής άσκησης και σύνδεσης με άλλες εφαρμογές και συσκευές fitness.



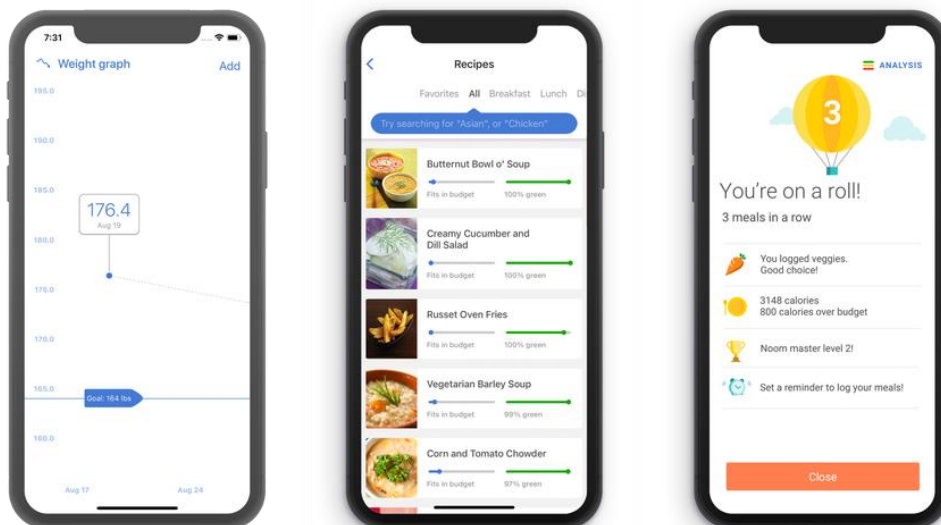
Εικόνα 1. Εφαρμογή MyFitnessPal

- **Lose It!:** Παρόμοια με το MyFitnessPal, αυτή η εφαρμογή επικεντρώνεται στην απώλεια βάρους μέσω της καταγραφής θερμίδων και τροφίμων. Οι χρήστες μπορούν να θέσουν στόχους απώλειας βάρους και να παρακολουθούν την πρόοδό τους.



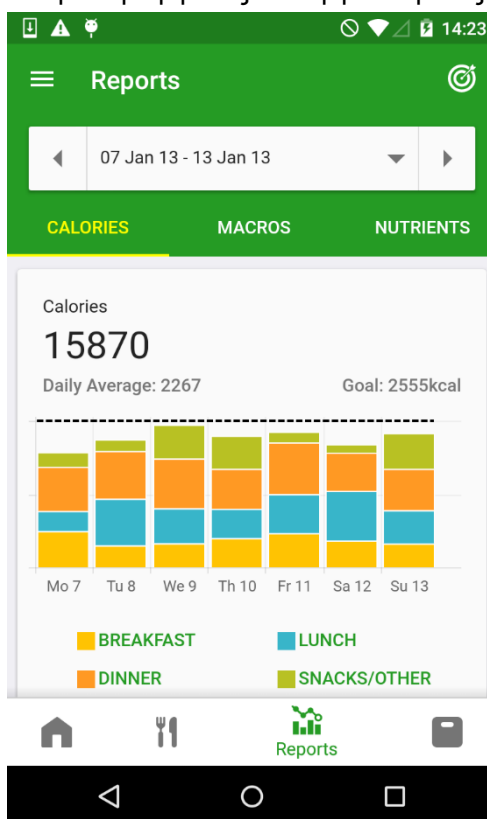
Εικόνα 2. Εφαρμογή Lose It!

- **Noom:** Η Noom ξεχωρίζει με την προσέγγισή της που βασίζεται στην ψυχολογία και την αλλαγή συμπεριφοράς. Εκτός από την παρακολούθηση της διατροφής και της άσκησης, προσφέρει καθοδήγηση και υποστήριξη από coaches και εστιάζει στην αλλαγή των διατροφικών συνηθειών.



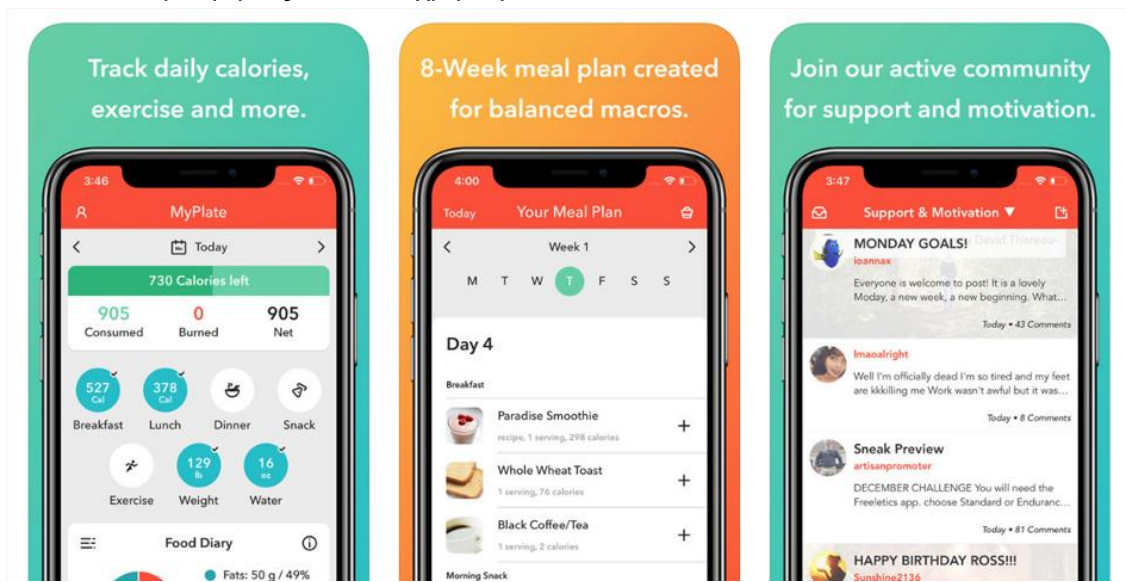
Εικόνα 3. Εφαρμογή Noom

- **FatSecret:** Μια εφαρμογή που προσφέρει λεπτομερείς πληροφορίες για τρόφιμα, ασκήσεις και συνταγές, επιτρέποντας στους χρήστες να παρακολουθούν τη διατροφή τους και τη φυσική τους δραστηριότητα.



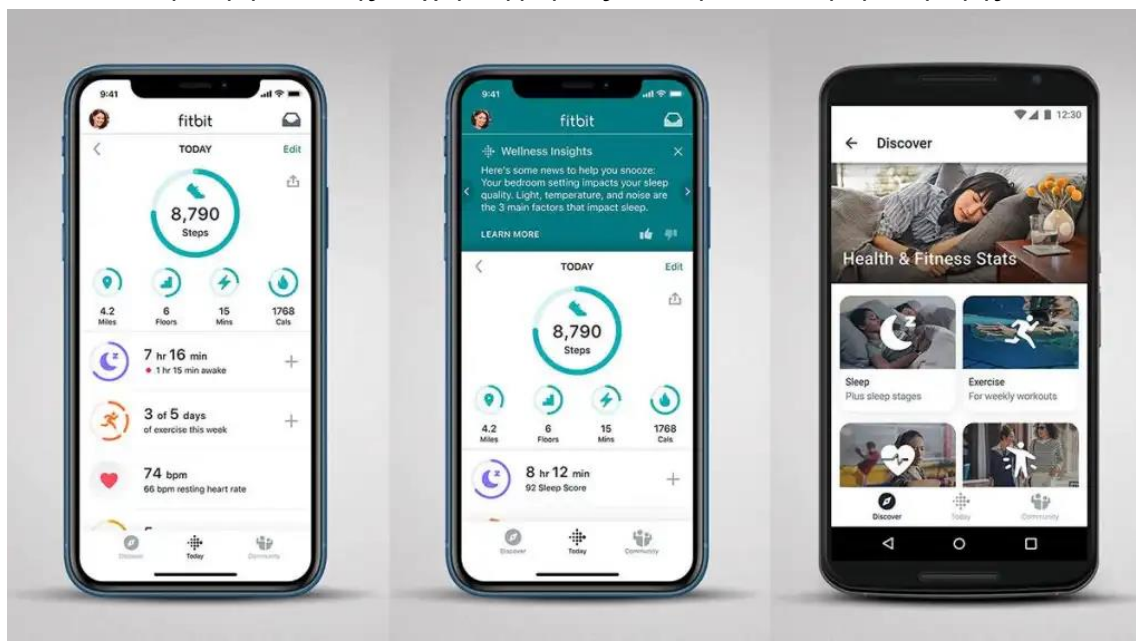
Εικόνα 4. Εφαρμογή FatSecret

- **MyPlate Calorie Tracker:** Μια εφαρμογή που προσφέρει εξατομικευμένες συμβουλές διατροφής και άσκησης, βασισμένη στους στόχους και τις προτιμήσεις του κάθε χρήστη.



Εικόνα 5. Εφαρμογή MyPlate Calorie Tracker

- **Fitbit:** Πέρα από την καταγραφή της φυσικής δραστηριότητας, η εφαρμογή προσφέρει επίσης διαχείριση βάρους και παρακολούθηση διατροφής.



Εικόνα 6. Εφαρμογή Fitbit

Παρά την ευρεία χρήση αυτών των εφαρμογών, υπάρχουν αρκετοί περιορισμοί που παραμένουν. Πολλές από αυτές εστιάζουν κυρίως στην καταγραφή θερμίδων και όχι στην παροχή εξατομικευμένων προτάσεων.

2.2. Επιστημονικές Βάσεις

Η δημιουργία μιας εφαρμογής για τη διαχείριση της διατροφής και της άσκησης βασίζεται σε επιστημονικές αρχές και έρευνες που καλύπτουν θέματα όπως οι διατροφικές ανάγκες, ο υπολογισμός θερμίδων, η σημασία της άσκησης και η εξατομίκευση των προτάσεων διατροφής. Η ενότητα αυτή θα εξετάσει τις βασικές θεωρητικές και πρακτικές γνώσεις που υποστηρίζουν την ανάπτυξη της εφαρμογής.

2.2.1. Διατροφικές Ανάγκες και Θερμίδες

Οι διατροφικές ανάγκες διαφέρουν ανάλογα με διάφορους παράγοντες, όπως το φύλο, η ηλικία, το βάρος, το ύψος και το επίπεδο φυσικής δραστηριότητας. Σύμφωνα με το Υπουργείο Υγείας και Κοινωνικής Αλληλεγγύης (2012), οι ημερήσιες ενεργειακές ανάγκες διαφέρουν ανάλογα με τα δημογραφικά χαρακτηριστικά και τον τρόπο ζωής. Για παράδειγμα, μια ενήλικη γυναίκα χρειάζεται κατά μέσο όρο 1800-2400 θερμίδες ημερησίως, ενώ ένας ενήλικος άνδρας χρειάζεται 2200-3000 θερμίδες ημερησίως.

Οι συνιστώμενες ημερήσιες θερμίδες μπορούν να υπολογιστούν με τη χρήση διαφόρων εξισώσεων, όπως η εξίσωση Harris-Benedict ή η πιο σύγχρονη εξίσωση Mifflin-St Jeor (Mifflin et al., 1990). Στην ελληνική βιβλιογραφία, ο Μπεχράκης (2002) αναφέρει ότι οι ημερήσιες ανάγκες σε θερμίδες μπορούν να επηρεαστούν σημαντικά από τον τρόπο ζωής και τις συνήθειες διατροφής του ατόμου¹⁰.

Είναι σημαντικό να κατανοήσουμε ότι ο υπολογισμός των ημερήσιων θερμίδων είναι κρίσιμος για τη διατήρηση της υγείας, καθώς η υπερβολική κατανάλωση θερμίδων μπορεί να οδηγήσει σε παχυσαρκία και σχετικές ασθένειες, όπως καρδιοπάθειες, διαβήτη τύπου 2 και υπέρταση ενώ αντίθετα, η υποκατανάλωση θερμίδων μπορεί να προκαλέσει απώλεια βάρους και ανεπάρκειες θρεπτικών συστατικών.¹¹

2.2.2. Υπολογισμός Ημερήσιας Κατανάλωσης Θερμίδων

Η μέτρηση των θερμίδων που καταναλώνει και καίει το σώμα είναι ένας σημαντικός παράγοντας για τη διατήρηση του ιδανικού βάρους. Ο Βασικός Μεταβολικός Ρυθμός (BMR) είναι η ποσότητα ενέργειας που χρειάζεται το σώμα σε κατάσταση ηρεμίας για να εκτελεί βασικές λειτουργίες όπως η αναπνοή, η κυκλοφορία του αίματος και η ρύθμιση της θερμοκρασίας. Οι επικρατέστεροι τύποι για τον υπολογισμό των ημερήσιων θερμίδων είναι οι τύποι των Harris-Benedict και Mifflin-St Jeor.

Ο αρχικός τύπος των Harris-Benedict δημιουργήθηκε το 1919 μετά από μελέτη των James Arthur Harris και Francis Gano Benedict. Οι εξισώσεις αναθεωρήθηκαν το 1984

¹⁰ Μπεχράκης, Π. (2002). Διατροφή και Υγεία. Εκδόσεις Παπαζήση.

¹¹ Υπουργείο Υγείας και Κοινωνικής Αλληλεγγύης. (2012). "Διατροφικές Οδηγίες για Ενήλικες". Αθήνα.

χρησιμοποιώντας νέα δεδομένα, προκειμένου να βελτιωθεί η ακρίβεια¹². Η αναθεωρημένη εξίσωση συνέχισε να χρησιμοποιείται μέχρι το 1990, όταν δημοσιεύτηκε η εξίσωση Mifflin-St Jeor, η οποία προτάθηκε ως μια πιο ακριβής μέθοδος υπολογισμού του BMR¹³.

- Πρωτότυπη εξίσωση Harris-Benedict (1919)
 - Για άνδρες: $BMR = 66,5 + (13,75 \times \text{βάρος σε κιλά}) + (5,003 \times \text{ύψος σε εκατοστά}) - (6,755 \times \text{ηλικία σε έτη})$
 - Για γυναίκες: $BMR = 655,1 + (9,563 \times \text{βάρος σε κιλά}) + (1,850 \times \text{ύψος σε εκατοστά}) - (4,676 \times \text{ηλικία σε έτη})$
- Αναθεωρημένη εξίσωση Harris-Benedict (1984)
 - Για άνδρες: $BMR = 88,362 + (13,397 \times \text{βάρος σε κιλά}) + (4,799 \times \text{ύψος σε εκατοστά}) - (5,677 \times \text{ηλικία σε έτη})$
 - Για γυναίκες: $BMR = 447,593 + (9,247 \times \text{βάρος σε κιλά}) + (3,098 \times \text{ύψος σε εκατοστά}) - (4,330 \times \text{ηλικία σε έτη})$
- Εξίσωση Mifflin-St Jeor (1990)
 - Για άνδρες: $BMR = (10 \times \text{βάρος σε κιλά}) + (6,25 \times \text{ύψος σε εκατοστά}) - (5 \times \text{ηλικία σε έτη}) + 5$
 - Για γυναίκες: $BMR = (10 \times \text{βάρος σε κιλά}) + (6,25 \times \text{ύψος σε εκατοστά}) - (5 \times \text{ηλικία σε έτη}) - 161$

Η συνολική κατανάλωση ενέργειας, γνωστή ως TDEE (Συνολική Ημερήσια Ενεργειακή Δαπάνη) υπολογίζεται προσθέτοντας την ενέργεια που καίγεται μέσω φυσικής δραστηριότητας στο BMR. Ο υπολογισμός αυτός λαμβάνει υπόψη διάφορα επίπεδα φυσικής δραστηριότητας, από καθιστική ζωή έως έντονη σωματική δραστηριότητα¹⁴.

2.2.3. Σημασία της Άσκησης

Η άσκηση είναι καθοριστικός παράγοντας για τη βελτίωση της συνολικής υγείας και τη διαχείριση του βάρους. Σύμφωνα με τις κατευθυντήριες γραμμές του Παγκόσμιου Οργανισμού Υγείας (WHO), οι ενήλικες θα πρέπει να συμμετέχουν σε τουλάχιστον 150 λεπτά μέτριας έντασης αερόβιας άσκησης ή 75 λεπτά έντονης έντασης αερόβιας άσκησης την εβδομάδα, σε συνδυασμό με ασκήσεις ενδυνάμωσης μυών δύο ή περισσότερες φορές την εβδομάδα. Η τακτική άσκηση συμβάλλει στη βελτίωση της καρδιοαναπνευστικής υγείας, στη μείωση του κινδύνου χρόνιων ασθενειών, στη διατήρηση της μυϊκής μάζας και της οστικής πυκνότητας, και στη βελτίωση της ψυχικής υγείας. Επίσης, η άσκηση μπορεί να αυξήσει την ευαισθησία στην ινσουλίνη και να συμβάλλει στη ρύθμιση των επιπέδων σακχάρου στο αίμα, γεγονός που είναι ιδιαίτερα σημαντικό για τα άτομα με διαβήτη τύπου 2.¹⁵

¹²Roza, A. M., & Shizgal, H. M. (1984). The Harris Benedict equation reevaluated: resting energy requirements and the body cell mass. *The American Journal of Clinical Nutrition*, 40(1), 168-182. doi:10.1093/ajcn/40.1.168

¹³Mifflin, M. D., St Jeor, S. T., Hill, L. A., Scott, B. J., Daugherty, S. A., & Koh, Y. O. (1990). A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition*, 51(2), 241-247. doi:10.1093/ajcn/51.2.241

¹⁴ Πάππας, Γ., & Κοτζαμάνης, Β. (2016). "Βασικός Μεταβολικός Ρυθμός και Θερμίδες". *Περιοδικό Κλινικής Διατροφής*, 12(1), 34-45.

¹⁵ Παγκόσμιος Οργανισμός Υγείας. (2015). "Κατευθυντήριες Γραμμές για τη Σωματική Δραστηριότητα". Γενεύη.

2.2.4. Εξατομικευμένες Προτάσεις Διατροφής

Οι εξατομικευμένες προτάσεις διατροφής βασίζονται σε δεδομένα όπως οι διατροφικές ανάγκες, οι προτιμήσεις, οι αλλεργίες και οι διατροφικοί περιορισμοί του χρήστη. Με τη χρήση προηγμένων αλγορίθμων και τεχνικών μηχανικής μάθησης, οι εφαρμογές μπορούν να προσαρμόσουν τις προτάσεις γευμάτων ώστε να ανταποκρίνονται στις συγκεκριμένες ανάγκες του κάθε χρήστη.

Οι εξατομικευμένες προτάσεις διατροφής έχουν αποδειχθεί ότι βελτιώνουν τη συμμόρφωση των ατόμων με υγιεινές διατροφικές συνήθειες. Με την παροχή προσαρμοσμένων πληροφοριών και προτάσεων, οι χρήστες είναι πιο πιθανό να ακολουθήσουν μια ισορροπημένη διατροφή και να επιτύχουν τους διατροφικούς τους στόχους¹⁶.

2.2.5. Συνεργασία Διαιτολόγων και Εφαρμογών

Η συνεργασία διαιτολόγων με εφαρμογές διατροφής και άσκησης μπορεί να προσφέρει προστιθέμενη αξία στους χρήστες. Οι διαιτολόγοι μπορούν να χρησιμοποιήσουν τα δεδομένα που συλλέγονται από τις εφαρμογές για να παρέχουν πιο στοχευμένες και αποτελεσματικές συμβουλές στους πελάτες τους. Επίσης, οι εφαρμογές μπορούν να λειτουργήσουν ως ένα επιπλέον εργαλείο παρακολούθησης και υποστήριξης για τους διαιτολόγους.

Η ενσωμάτωση των επαγγελματικών συμβουλών των διαιτολόγων σε εφαρμογές μπορεί να ενισχύσει την αξιοπιστία και την αποτελεσματικότητα των εφαρμογών. Οι χρήστες μπορούν να λάβουν εξατομικευμένες συμβουλές και να παρακολουθούν την πρόοδό τους σε πραγματικό χρόνο, ενώ οι διαιτολόγοι μπορούν να αξιοποιήσουν τα δεδομένα για να παρέχουν καλύτερη καθοδήγηση.¹⁷

2.2.6. Τεχνολογίες Παρακολούθησης Υγείας

Η χρήση φορητών συσκευών παρακολούθησης υγείας, όπως τα βραχιόλια και τα smartwatches, έχει αυξηθεί σημαντικά τα τελευταία χρόνια. Αυτές οι συσκευές μπορούν να παρακολουθούν τη φυσική δραστηριότητα, τις καύσεις θερμίδων, τον ύπνο και άλλες παραμέτρους υγείας, παρέχοντας πολύτιμα δεδομένα στους χρήστες και στους διαιτολόγους τους.

Οι φορητές συσκευές μπορούν να συνδεθούν με εφαρμογές διατροφής και άσκησης, επιτρέποντας την αυτόματη καταγραφή και ανάλυση των δεδομένων. Αυτή η ολοκληρωμένη προσέγγιση μπορεί να βοηθήσει τους χρήστες να διαχειριστούν καλύτερα την υγεία τους και να επιτύχουν τους στόχους τους πιο αποτελεσματικά¹⁸.

¹⁶ Τμήμα Διαιτολογίας και Διατροφής, Χαροκόπειο Πανεπιστήμιο. (2020). "Εξατομικευμένες Διατροφικές Προτάσεις και Συμμόρφωση". Αθήνα.

¹⁷ Ακαδημία Διατροφής και Διαιτολογίας. (2017). "Η σημασία της συνεργασίας διαιτολόγων και τεχνολογίας". Περιοδικό Διατροφικής Θεραπείας, 15(2), 56-67.

¹⁸ Ινστιτούτο Τεχνολογίας Υγείας. (2021). "Η χρήση φορητών συσκευών στην παρακολούθηση της υγείας". Τεχνολογία Υγείας, 18(3), 78-89.

3. Μεθοδολογία

3.1. Ανάλυση Απαιτήσεων

Στην ανάλυση απαιτήσεων, θα περιγράψουμε τις λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής.

3.1.1. Λειτουργικές Απαιτήσεις

Η εφαρμογή περιλαμβάνει πολλές λειτουργίες που καλύπτουν τις ανάγκες των χρηστών σε διάφορες πτυχές της διατροφής και της άσκησης. Πρώτα, οι χρήστες εισάγουν προσωπικά δεδομένα, όπως η ηλικία, το ύψος, το βάρος, το φύλο, το επίπεδο φυσικής δραστηριότητας και τον στόχο τους (π.χ., απώλεια βάρους, αύξηση μυϊκής μάζας, ή διατήρηση του βάρους). Με βάση αυτά τα δεδομένα, η εφαρμογή υπολογίζει τις ημερήσιες θερμίδες που χρειάζεται ο χρήστης. Οι χρήστες μπορούν να αναζητήσουν τρόφιμα και να δουν τις θερμίδες και τα διατροφικά στοιχεία τους. Παρέχεται επίσης λίστα ασκήσεων με τον υπολογισμό των θερμίδων που καίγονται και σύνδεσμοι σε βίντεο οδηγιών στο YouTube. Η εφαρμογή προσφέρει προγραμματιστή γευμάτων με συνταγές, οι οποίες είναι διαχωρισμένες ανάλογα με το είδος του γεύματος (πρωινό, μεσημεριανό, βραδινό). Οι χρήστες μπορούν να εγγραφούν και να εισέλθουν μέσω Facebook, Google, email ή ως ανώνυμοι χρήστες, ενώ υπάρχουν επίσης υπενθυμίσεις για την κατανάλωση νερού.

3.1.2. Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις της εφαρμογής περιλαμβάνουν διάφορες παραμέτρους που διασφαλίζουν την ομαλή λειτουργία και την ασφάλεια της εφαρμογής. Απαιτείται η εφαρμογή να ανταποκρίνεται γρήγορα στις εντολές των χρηστών, με χρόνο απόκρισης λιγότερο από δύο δευτερόλεπτα για τις περισσότερες λειτουργίες και να είναι διαθέσιμη το 99.9% του χρόνου. Στον τομέα της ασφάλειας, η εφαρμογή προστατεύει τα προσωπικά δεδομένα των χρηστών μέσω κρυπτογράφησης και αυθεντικοποίησης μέσω Facebook, Google ή email, με τα δεδομένα να αποθηκεύονται με ασφάλεια στη Firebase. Η ευχρηστία είναι επίσης κρίσιμη, καθώς η εφαρμογή θα πρέπει να έχει εύχρηστη διεπαφή χρήστη, εύκολη στην πλοήγηση και κατανοητή ακόμα και για μη εξοικειωμένους χρήστες. Τέλος, η επεκτασιμότητα είναι απαραίτητη, ώστε η εφαρμογή να μπορεί να υποστηρίξει την αύξηση του αριθμού χρηστών και την προσθήκη νέων λειτουργιών στο μέλλον.

3.2. Σχεδιασμός Συστήματος

Παρακάτω θα παρατεθεί ο σχεδιασμός της αρχιτεκτονικής της εφαρμογής, οι τεχνολογίες που χρησιμοποιήθηκαν και η βάση δεδομένων.

3.2.1. Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική της εφαρμογής ακολουθεί το μοντέλο client-server, διασφαλίζοντας ότι η εφαρμογή στο κινητό τηλέφωνο λειτουργεί ως ο client και συνδέεται με servers για την αποθήκευση δεδομένων και την εκτέλεση λειτουργιών. Στην πλευρά του client, η εφαρμογή Android είναι γραμμένη σε Java και αναπτύχθηκε χρησιμοποιώντας το Android Studio. Στην πλευρά του server, η αποθήκευση δεδομένων και οι λειτουργίες backend υποστηρίζονται από τη Firebase. Αυτή η αρχιτεκτονική επιτρέπει την αξιόπιστη και ασφαλή διαχείριση των δεδομένων, καθώς και την επεκτασιμότητα για μελλοντικές αναβαθμίσεις και βελτιώσεις.

3.2.2. Τεχνολογίες και Εργαλεία

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν διάφορες τεχνολογίες και εργαλεία, που διασφαλίζουν την αποτελεσματική και αξιόπιστη λειτουργία της. Η κύρια γλώσσα προγραμματισμού είναι η Java, που χρησιμοποιείται για την ανάπτυξη της εφαρμογής Android. Η πλατφόρμα ανάπτυξης που χρησιμοποιήθηκε είναι το Android Studio, το οποίο προσφέρει ένα ολοκληρωμένο περιβάλλον για τη δημιουργία και τη δοκιμή της εφαρμογής. Η βάση δεδομένων που επιλέχθηκε είναι η Firebase, η οποία παρέχει δυνατότητες αποθήκευσης δεδομένων σε πραγματικό χρόνο. Επιπλέον, χρησιμοποιήθηκαν διάφορα APIs για την απόκτηση δεδομένων τροφίμων και συνταγών, όπως το CalorieNinjas API, το Spoonacular API και το YouTube API που παρέχουν αξιόπιστες και ενημερωμένες πληροφορίες.

3.2.3. Δομή της Βάσης Δεδομένων

Η βάση δεδομένων της εφαρμογής χρησιμοποιεί τη Firebase, μια βάση δεδομένων NoSQL που επιτρέπει την αποθήκευση και ανάκτηση δεδομένων σε πραγματικό χρόνο. Η δομή της βάσης δεδομένων είναι σχεδιασμένη έτσι ώστε να καλύπτει τις ανάγκες της εφαρμογής με αποτελεσματικό τρόπο. Περιλαμβάνει συλλογή χρηστών, όπου αποθηκεύονται δεδομένα όπως ηλικία, ύψος, βάρος, φύλο, επίπεδο φυσικής δραστηριότητας και στόχος. Τέλος, υπάρχει συλλογή ασκήσεων που περιλαμβάνει δεδομένα για τις ασκήσεις, όπως θερμίδες που καίγονται και σύνδεσμοι για βίντεο στο YouTube.

3.2.4. Άλλες Τεχνολογίες και Εργαλεία

Εκτός από τις κύριες τεχνολογίες, χρησιμοποιήθηκαν και άλλα εργαλεία για την ανάπτυξη και παρακολούθηση της εφαρμογής. Το Git χρησιμοποιήθηκε για τον έλεγχο εκδόσεων και τη συνεργασία στην ανάπτυξη του κώδικα, διασφαλίζοντας ότι όλες οι αλλαγές παρακολουθούνται και μπορούν να αναστραφούν εάν χρειαστεί. Το Google Analytics χρησιμοποιήθηκε για την παρακολούθηση της χρήσης της εφαρμογής και την ανάλυση της συμπεριφοράς των χρηστών, παρέχοντας πολύτιμες πληροφορίες για τη βελτίωση της εφαρμογής. Αυτά τα εργαλεία συμβάλλουν στη διασφάλιση της ποιότητας και της αποδοτικότητας της ανάπτυξης και της λειτουργίας της εφαρμογής.

3.3. Περιγραφή Λειτουργιών

Η εφαρμογή προσφέρει μια εκτεταμένη σειρά λειτουργιών που καλύπτουν όλες τις ανάγκες των χρηστών στους τομείς της διατροφής και της άσκησης. Ακολουθεί μια αναλυτική περιγραφή των κυριότερων λειτουργιών:

1. **Εισαγωγή Προσωπικών Δεδομένων:** Οι χρήστες εισάγουν τα προσωπικά τους δεδομένα, όπως ηλικία, ύψος, βάρος, φύλο, επίπεδο φυσικής δραστηριότητας και στόχος (απώλεια βάρους, αύξηση μυϊκής μάζας, διατήρηση βάρους).
2. **Υπολογισμός Ημερήσιων Θερμίδων:** Η εφαρμογή υπολογίζει τις ημερήσιες θερμίδες που απαιτούνται από τον κάθε χρήστη βάσει των εισαγμένων δεδομένων.
3. **Καταγραφή Κατανάλωσης Τροφών:** Οι χρήστες μπορούν να αναζητούν τρόφιμα, να επιλέγουν την ποσότητα που καταναλώνουν και να καταγράφουν την κατανάλωσή τους, επιτρέποντας έτσι την παρακολούθηση των θερμίδων που καταναλώνουν.
4. **Προτάσεις Γυμναστικής:** Η εφαρμογή παρέχει μια λίστα ασκήσεων, με κάθε άσκηση να συνοδεύεται από ένα tutorial της άσκησης στο YouTube για την εκτέλεσή τους.
5. **Προγραμματισμός Γευμάτων:** Οι χρήστες μπορούν να δημιουργήσουν προσαρμοσμένα προγράμματα γευμάτων βάσει των προτιμήσεών τους και των θερμίδων που έχουν διαθέσιμες.
6. **Συνταγές:** Η εφαρμογή παρέχει μια ποικιλία συνταγών για πρωινό, μεσημεριανό και βραδινό, οι οποίες προσαρμόζονται στις διατροφικές ανάγκες των χρηστών.

4. Ανάπτυξη Εφαρμογής

4.1. Περιγραφή Ανάπτυξης

Η διαδικασία ανάπτυξης της εφαρμογής ακολούθησε ένα προσεκτικά σχεδιασμένο σχέδιο βήμα προς βήμα, ξεκινώντας από την ανάλυση των λειτουργικών απαιτήσεων. Έγινε μια συλλογή των λειτουργικών και μη λειτουργικών απαιτήσεων, με την προσδιορισμό των βασικών χαρακτηριστικών της εφαρμογής, συμπεριλαμβανομένων των λειτουργιών εισαγωγής προσωπικών δεδομένων, της αναζήτησης τροφίμων και ασκήσεων, καθώς και της δημιουργίας συνταγών και προγραμμάτων γευμάτων.

Στη συνέχεια, προχωρήσαμε στον σχεδιασμό της αρχιτεκτονικής της εφαρμογής, χρησιμοποιώντας τεχνολογίες όπως η Java για τον κώδικα της εφαρμογής, το Android Studio για τον εξομοιωτή και το Firebase ως βάση δεδομένων. Επιπλέον, χρησιμοποιήθηκαν διάφορα API, όπως τα `api.calorieninjas.com`, `api.youtube.com` και `api.spoonacular.com`, για τη λήψη δεδομένων σχετικά με τις θρεπτικές αξίες των τροφίμων και τις ασκήσεις.

Αυτή η φάση περιελάμβανε μια λεπτομερή ανάλυση του ρόλου των προσωπικών δεδομένων του χρήστη στη λειτουργία της εφαρμογής. Ενώ η αυθεντικοποίηση του χρήστη γίνεται μέσω της Firebase για την εξασφάλιση της ασφάλειας, η αποθήκευση των προσωπικών δεδομένων του χρήστη γίνεται τοπικά στις `SharedPreferences` της συσκευής.

Η υλοποίηση της λειτουργικότητας εισαγωγής των δεδομένων του χρήστη πραγματοποιήθηκε με έμφαση στην ευκολία χρήσης και την αποτελεσματική διαχείριση των πληροφοριών. Ο χρήστης προσκλήθηκε να εισάγει τα δεδομένα του, όπως η ηλικία, το βάρος και άλλες σχετικές παραμέτρους μέσω μιας φιλικής προς τον χρήστη διεπαφής. Τα δεδομένα αυτά στη συνέχεια αποθηκεύτηκαν στις `SharedPreferences` της συσκευής, εξασφαλίζοντας τη διατήρησή τους ακόμη και όταν ο χρήστης είναι εκτός σύνδεσης.

Αυτή η προσέγγιση επιτρέπει στους χρήστες να διατηρούν τις προσωπικές τους πληροφορίες ακόμη και χωρίς σύνδεση στο διαδίκτυο, παρέχοντας τους ένα αίσθημα ασφάλειας και ιδιωτικότητας. Επιπλέον, η αυτόνομη αποθήκευση των δεδομένων στη συσκευή μειώνει την εξάρτηση από εξωτερικές υπηρεσίες και εξασφαλίζει τη συνεχή πρόσβαση σε αυτά, ανεξαρτήτως της κατάστασης της σύνδεσης στο διαδίκτυο.

4.2. Τεχνολογίες και Εργαλεία

Η ανάπτυξη της εφαρμογής βασίστηκε σε μια ποικιλία τεχνολογιών και εργαλείων που επέτρεψαν την ομαλή λειτουργία και τη δημιουργία μιας ευχάριστης εμπειρίας χρήστη. Η βασική γλώσσα προγραμματισμού που χρησιμοποιήθηκε ήταν η Java, η οποία προσφέρει μεγάλη ευελιξία και αξιοπιστία στην ανάπτυξη εφαρμογών Android.

Για τη διαχείριση της βάσης δεδομένων, επιλέχθηκε η Firebase για την εξαιρετική της ευελιξία και τη δυνατότητά της να προσφέρει υψηλή ασφάλεια και αξιοπιστία. Επιπλέον, για την ανάλυση της διατροφής και των θερμίδων των τροφίμων, χρησιμοποιήθηκαν διάφορα API όπως το API του `CalorieNinjas` και το `Spoonacular API`, ενώ για την προβολή tutorial ασκήσεων, χρησιμοποιήθηκε το `Youtube Api`. Αυτές οι πηγές δεδομένων βοήθησαν στην παροχή αξιόπιστων πληροφοριών σχετικά με τις τροφές και τη διατροφή στους χρήστες.

Για τον σχεδιασμό της διεπαφής χρήστη (UI), χρησιμοποιήθηκε το `Android Studio`, προσφέροντας έναν εύκολο τρόπο δημιουργίας και επεξεργασίας των γραφικών στοιχείων

της εφαρμογής. Η χρήση του Android Studio επέτρεψε την ανάπτυξη μιας ευέλικτης και αποδοτικής διεπαφής που προσφέρει μια φιλική προς τον χρήστη εμπειρία.

Συνολικά, η χρήση αυτών των τεχνολογιών και εργαλείων συνέβαλε στη δημιουργία μιας υψηλής ποιότητας εφαρμογής που προσφέρει αξιόπιστες πληροφορίες και μια ευχάριστη εμπειρία χρήστη.

5. Ανάλυση της Εφαρμογής

5.1. Εγκατάσταση και Ρυθμίσεις:

Η εφαρμογή Fitdiet αναπτύχθηκε μέσω του Android Studio IDE. Η εγκατάσταση και οι αρχικές ρυθμίσεις της εφαρμογής FitDiet απαιτούν την τήρηση ορισμένων διαδικασιών για να εξασφαλιστεί η ομαλή λειτουργία της στις συσκευές Android και iOS.

Για τη σωστή εγκατάσταση και λειτουργία της εφαρμογής, απαιτείται η ύπαρξη ενός λειτουργικού συστήματος Android με έκδοση 4.4 (KitKat) και άνω. Επιπλέον, προτείνεται η διαθεσιμότητα ελεύθερου χώρου αποθήκευσης τουλάχιστον 100MB για την εφαρμογή. Για την εύρυθμη λειτουργία της εφαρμογής χρειάζεται ο παραπάνω χώρος καθώς ενδέχεται να αποθηκευτούν εικόνες, video, και δεδομένα του χρήστη, προσωπικές ρυθμίσεις και άλλα αρχεία που χρησιμοποιεί κατά τη λειτουργία της.

Η εφαρμογή αποτελείται από 16 οθόνες (activities/fragments) και 42 java classes, οι οποίες χρησιμοποιούνται είτε ως adapters για τη δημιουργία λιστών είτε ως κατηγορία αντικειμένων. Οι κυριότερες οθόνες που αναπτύχθηκαν είναι:

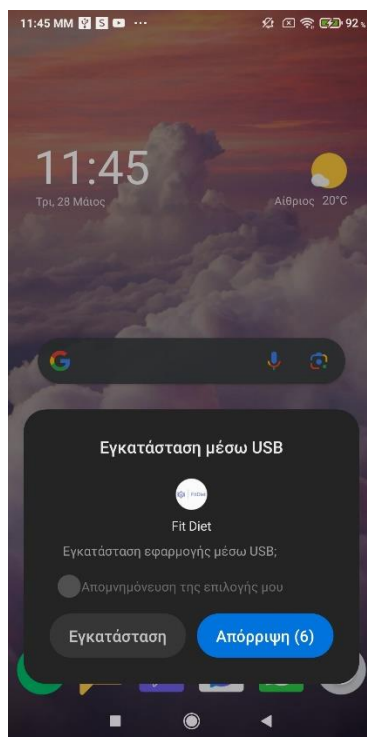
- **SplashActivity:** Αναπτύχθηκε για την εμφάνιση μιας εικόνας εκκίνησης στον χρήστη κατά την εκκίνηση της εφαρμογής.
- **FacebookActivity:** Είναι η πρώτη οθόνη που συναντά ο χρήστης και μπορεί να συνδεθεί ή να εγγραφεί στην εφαρμογή μέσω Facebook, Google, email ή με ανώνυμη σύνδεση.
- **MainActivity:** Αποτελεί την κύρια οθόνη και τον κεντρικό χώρο πλοήγησης του χρήστη.
- **AboutFragment:** Αναπτύχθηκε για να παρουσιάσει πληροφορίες σχετικά με την εφαρμογή.
- **RemindersActivity:** Χρησιμοποιείται για τη διαχείριση των υπενθυμίσεων που λαμβάνει ο χρήστης για την κατανάλωση του νερού.
- **Fragment4:** Αναπτύχθηκε για να παρουσιάσει τα πεδία που έχουν υπολογιστεί βάσει των δεδομένων που έχει δώσει ο χρήστης στα Fragment1, Fragment2, Fragment3 & Fragment6, ήτοι Υπολογισμός και εμφάνιση ΔΜΣ, Ημερήσια κατανάλωση νερού και Ημερήσια κατανάλωση θερμίδων
- **Recipes:** Αναπτύχθηκε για να παρουσιάσει συνταγές βάσει το γεύματος που έχει επιλέξει ο χρήστης (πρωινό, μεσημεριανό ή βραδινό).
- **RecipeDetailsFragment:** Αναπτύχθηκε για να παρουσιάσει λεπτομέρειες για συνταγή που έχει επιλέξει ο χρήστης.
- **MealPlanFragment:** Αναπτύχθηκε για να εμφανίζει ένα ημερήσιο πλάνο γευμάτων και τα διατροφικά στοιχεία που περιέχει στο χρήστη.
- **ApiActivity:** Αναπτύχθηκε για να ανακτά και να προβάλλει στο χρήστη τα διατροφικά στοιχεία του φαγητού που έχει αναζητήσει ο χρήστης, καθώς και να προσθέτει ή να αφαιρεί τις θερμίδες από τον counter για τις ημερήσιες θερμίδες που έχει καταναλώσει ο χρήστης.
- **LearnActivity:** Είναι υπεύθυνο για την εμφάνιση των κατηγοριών ασκήσεων
- **Exercise:** Αναλαμβάνει την εμφάνιση μιας λίστας με βίντεο ασκήσεων από το YouTube βάσει της κατηγορίας που έχει επιλέξει ο χρήστης.
- **TutorialPlayback** Είναι υπεύθυνη για την αναπαραγωγή βίντεο από το YouTube και την εμφάνιση των πληροφοριών του video, ώστε να έχει περαιτέρω καθοδήγηση ο χρήστης

Επιπλέον, μερικές σημαντικές java classes είναι οι παρακάτω

- **RecipeApi/YoutubeAPI/CalorieNinjasApi:** Χρησιμοποιείται για την επικοινωνία με τα API της εφαρμογής.
- **FirestoreManager:** Χρησιμοποιείται για την καταχώρηση των στοιχείων του χρήστη (ηλικία, βάρος, ύψος κτλ.) στη Firebase και τους πίνακες UserData & UserMetrics.

5.2. Παρουσίαση και τεχνική ανάλυση της Εφαρμογής Fitdiet

Μετά την εγκατάσταση της εφαρμογής και κατά την εκκίνηση της εφαρμογής, η οθόνη εκκίνησης προβάλλει μια Splash Screen στον χρήστη.



Εικόνα 7. Εγκατάσταση Εφαρμογής Fitdiet

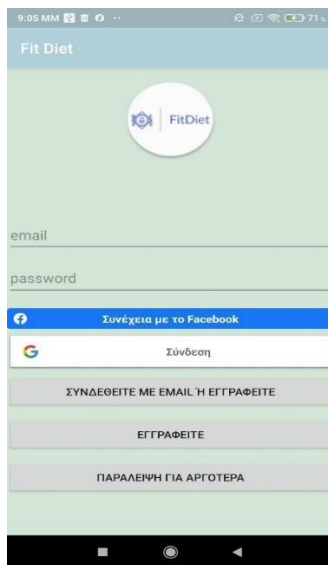


Εικόνα 8. Splash screen Εφαρμογής.

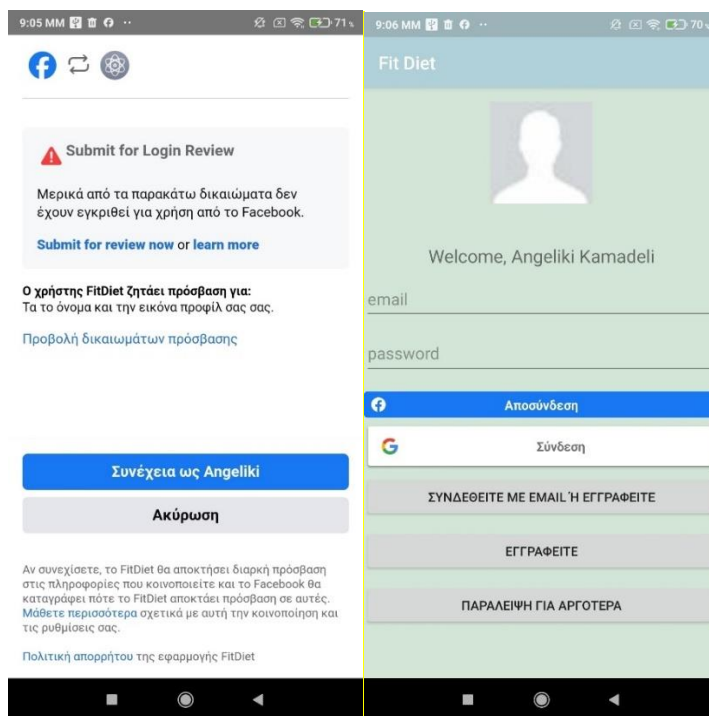
5.2.1. Οθόνη εγγραφής και σύνδεσης χρήστη - FacebookActivity

Η πρώτη βασική οθόνη με την οποία έρχεται σε επαφή ο χρήστης είναι αυτή που σχετίζεται με το Login ή Sign up του χρήστη. Η java class FacebookActivity χρησιμοποιεί το Firebase Authentication για την επαλήθευση χρηστών μέσω email και κωδικού πρόσβασης, ανώνυμης σύνδεσης, καθώς και σύνδεσης μέσω Facebook και Google. Σε περίπτωση που επιθυμεί να εγγραφεί με email μπορεί να επιλέξει το button «Εγγραφείτε».

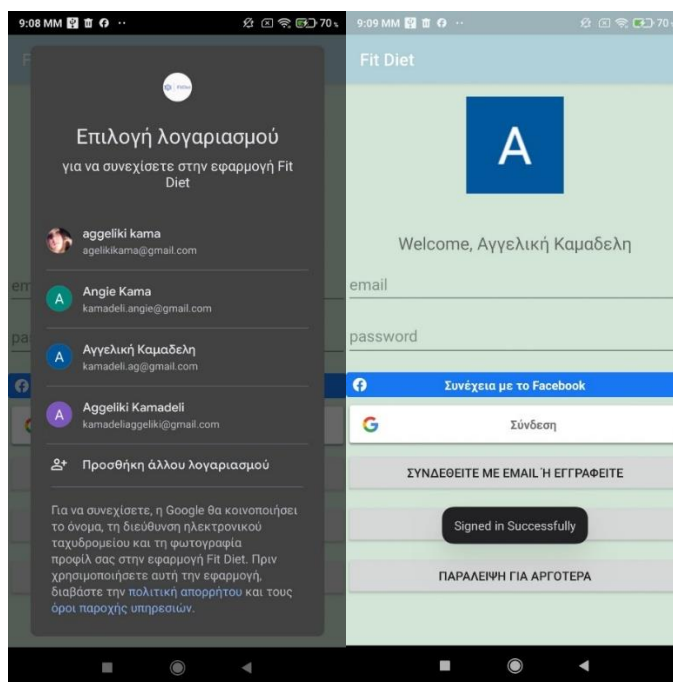
Η FacebookActivity «κληρονομεί» από την AppCompatActivity και υλοποιεί το View.OnClickListener. Αυτή η κλάση χρησιμοποιεί το Firebase Authentication για να επιτρέπει στους χρήστες να συνδέονται με διάφορες μεθόδους αυθεντικοποίησης.



Εικόνα 9. Οθόνη σύνδεσης με επιλογές εγγραφή και σύνδεση με email, Facebook, και Google και ανώνυμη σύνδεση (Παράλειψη για αργότερα).

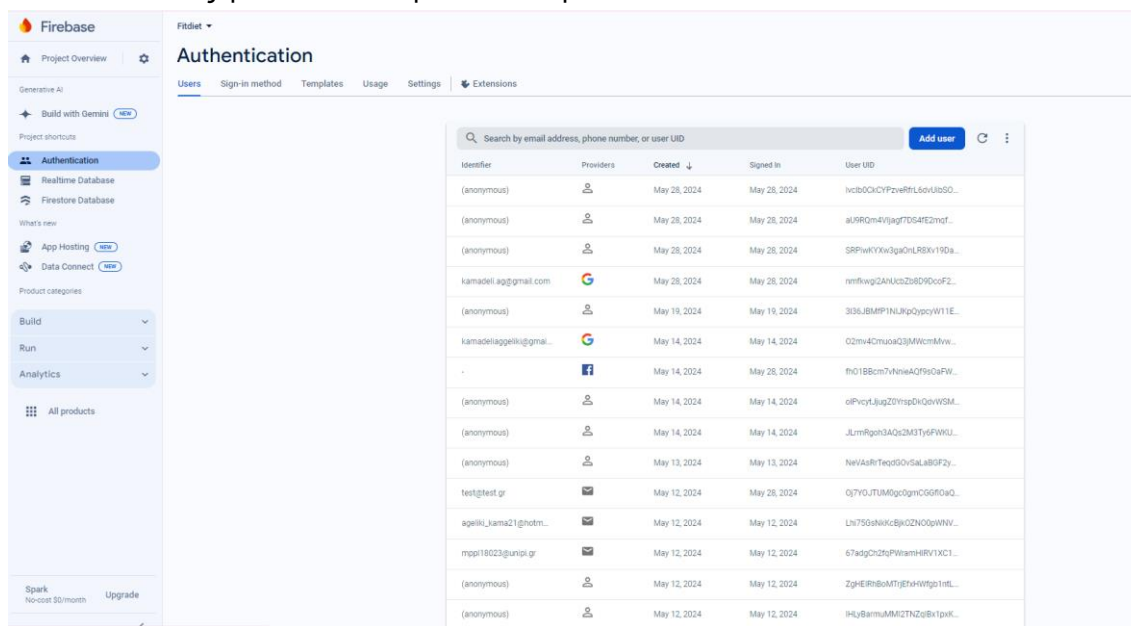


Εικόνες 10-11: Επιτυχής σύνδεση μέσω Facebook.



Εικόνες 12-13: Επιτυχημένη σύνδεση μέσω Google.

Μετά την επιτυχή εγγραφή ή σύνδεση υπάρχει η σχετική καταχώρηση στο Firebase Authentication όπως φαίνεται και στην εικόνα παρακάτω.



Εικόνα 14. Firebase Authentication.

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη FacebookActivity είναι:

- RC_SIGN_IN: Σταθερά για την κωδικοποίηση του αιτήματος σύνδεσης.
- Διάφορα UI στοιχεία όπως ImageView, TextView, EditText, Button, και άλλα.

Μέθοδοι

Αρχικά, η μέθοδος onCreate καλείται όταν η δραστηριότητα δημιουργείται στην οποία γίνεται αρχικοποίηση των στοιχείων UI και του Firebase Auth, ορίζεται το layout της δραστηριότητας και γίνεται η εγγραφή κλήσεων επιστροφής για τη σύνδεση του χρήστη με το Facebook και τη Google. Επιπλέον, πραγματοποιείται ο έλεγχος και η εμφάνιση μηνυμάτων σφάλματος στον κωδικό πρόσβασης αν είναι μικρότερος από 6 χαρακτήρες.

Η μέθοδος onClick διαχειρίζεται τα κλικ των κουμπιών για email login, ανώνυμη σύνδεση, Facebook login και Google login. Ανάλογα με το κουμπί που πατήθηκε, καλούνται οι αντίστοιχες μέθοδοι loginUserWithEmailAndPassword, signUpUserWithEmailAndPassword, signInAnonymously ή signIn.

Στις υπόλοιπες μεθόδους, γίνεται η διαχείριση των αποτελεσμάτων των δραστηριοτήτων που κλήθηκαν με startActivityForResult, συμπεριλαμβανομένων των δεδομένων από το Facebook και το Google sign-in (onActivityResult), της σύνδεσης μέσω Google (FirebaseGoogleAuth) ή Facebook (handleFacebookAccessToken) με τα διαπιστευτήρια του χρήστη για να συνδεθεί στη Firebase.

5.2.2. Interface επικοινωνίας των Fragments - FragmentsCommunicator

Για τις περισσότερες οθόνες έχουν χρησιμοποιηθεί Fragments και η επικοινωνία μεταξύ των οθονών επιτυγχάνεται με τη βοήθεια του interface FragmentsCommunicator Interface

Το FragmentsCommunicator είναι ένα public interface που παρέχει τρεις μεθόδους για την επικοινωνία μεταξύ των fragments. Η μέθοδος respond(String data1, int data2) επιτρέπει την αποστολή δεδομένων από ένα fragment σε άλλο. Η μέθοδος goToRecipeDetails(Recipe recipe) επιτρέπει τη μετάβαση στο fragment που δείχνει λεπτομέρειες μιας συνταγής. Η μέθοδος goToMealPlanFragment(String targetCalories) μεταφέρει το χρήστη στο fragment που αφορά τον προγραμματισμό γευμάτων, χρησιμοποιώντας τις θερμίδες στόχο ως παράμετρο.

5.2.3. Κύρια Οθόνη - MainActivity

Η MainActivity, είναι η Homepage της εφαρμογής και χρησιμοποιεί το interface FragmentsCommunicator, καθώς και την NavigationView.OnNavigationItemSelectedListener. Αρχικά, γίνεται η αρχικοποίηση του Firebase Authentication και του Realtime Database reference. Η μεταβλητή isLoggedIn χρησιμοποιείται για την αποθήκευση της κατάστασης σύνδεσης του χρήστη. Στην onCreate, γίνεται αρχικοποίηση του layout, του toolbar, και των στοιχείων UI, όπως το TextView και τα buttons login και logout.

Η πρώτη εικόνα που θα δει ο χρήστης μετά το login θα είναι η παρακάτω. Όπως φαίνεται μπορεί να επιλέξει να ξεκινήσει την ανάλυση σώματος, ήτοι να καταχωρήσει τα στοιχεία του (όπως φύλο, ηλικία, βάρος κτλ.) και να δει τις ημερήσιες θερμίδες που μπορεί να

καταναλώσει, τις ημερήσιες ανάγκες του σε νερό. Επιπλέον, μπορεί να επιλέξει ένα από τα buttons για πρωινό, μεσημεριανό ή βραδινό ώστε να δει υγιεινές συνταγές για τα γεύματα ή να επιλέξει ένα πλάνο διατροφής.



Εικόνα 15. Homepage.

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη MainActivity είναι:

- mAuth: Αντικείμενο FirebaseAuth για τη διαχείριση της αυθεντικοποίησης χρηστών.
- userRef: Αναφορά στη βάση δεδομένων Firebase για την αποθήκευση δεδομένων χρήστη.
- doubleBackToast: Toast μήνυμα για την επιβεβαίωση εξόδου από την εφαρμογή.
- Διάφορα UI στοιχεία όπως MenuItem, TextView, Button, και άλλα.
- signIn: Μεταβλητή String για την αποθήκευση του τρόπου σύνδεσης του χρήστη.
- weight, height, gender, goal, physical_activity: Μεταβλητές String για την αποθήκευση των στοιχείων του χρήστη.

Μέθοδοι

Η MainActivity περιλαμβάνει διάφορες μεθόδους που διαχειρίζονται την επικοινωνία και τη μετάβαση από το ένα fragment στο άλλο, την αποθήκευση και ανάκτηση δεδομένων χρήστη, καθώς και τις ενέργειες στο μενού και το navigation drawer. Η επικοινωνία με τα παραπάνω Fragments γίνεται με τη μέθοδο `respond(String data1, int data2)`, η οποία χειρίζεται την επικοινωνία μεταξύ των fragments, αποθηκεύοντας δεδομένα στο SharedPreferences και φορτώνοντας τα κατάλληλα fragments. Στη συνέχεια, η μέθοδος `loadFragment` φορτώνει ένα

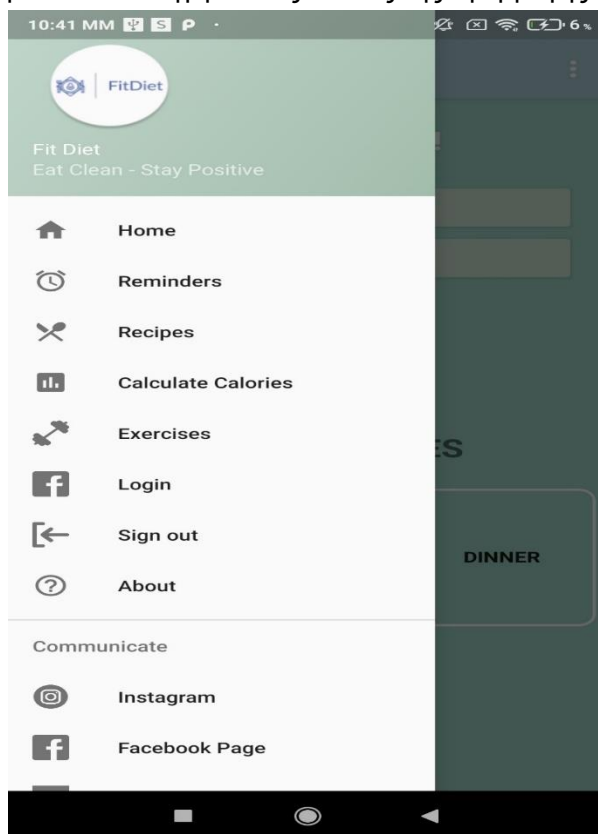
fragment στον καθορισμένο layout. Οι μέθοδοι `goToRecipeDetails` και `goToMealPlanFragment` χρησιμοποιούνται για να μεταφέρουν τον χρήστη στα αντίστοιχα fragments για τις συνταγές και το πλάνο διατροφής.

Η μέθοδος `saveData()` αποθηκεύει δεδομένα του χρήστη στο `SharedPreferences` και αντίστοιχα η μέθοδος `getMyData` ανακτά συγκεκριμένα δεδομένα από το `SharedPreferences` ανάλογα με την παράμετρο που δίνεται.

Η μέθοδος `onCreateOptionsMenu` φορτώνει το μενού στην action bar. Οι μέθοδοι `login` και `logout` διαχειρίζονται τη σύνδεση και την αποσύνδεση του χρήστη, ενημερώνοντας το μενού ανάλογα. Η μέθοδος `onOptionsItemSelected` χειρίζεται τις επιλογές του μενού, ενώ η `onNavigationItemSelected` διαχειρίζεται τα clicks στο navigation drawer, φορτώνοντας τα κατάλληλα fragments ή ξεκινώντας νέες δραστηριότητες.

Η `MainActivity` αρχικοποιεί τις υπηρεσίες του Firebase και τα στοιχεία UI κατά τη δημιουργία. Οι μέθοδοι του `FragmentsCommunicator` χρησιμοποιούνται για την επικοινωνία μεταξύ των fragments και τη φόρτωση των κατάλληλων δεδομένων ή UI elements. Οι μέθοδοι αποθήκευσης και ανάκτησης δεδομένων διασφαλίζουν ότι τα δεδομένα του χρήστη παραμένουν συνεπή και προσβάσιμα σε όλη την εφαρμογή.

Όταν ο χρήστης επιλέξει το menu (☰) θα δει στο navigation drawer (activity_main_drawer) τα fragment για να μπορεί να πλοηγηθεί στις οθόνες της εφαρμογής.



Εικόνα 16. Navigation drawer

Εάν ο χρήστης επιλέξει ένα από τα 3 items στο πεδίο «Επικοινωνία», θα μεταβεί στα social media (Instagram, Facebook) ή στο site του Πανεπιστημίου Πειραιά



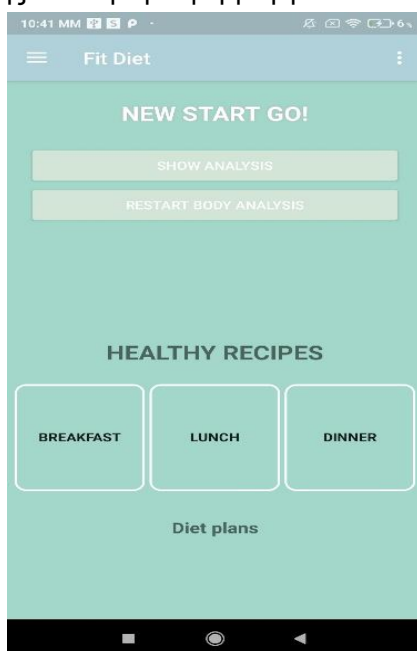
Εικόνα 17. Σελίδα Instagram Πανεπιστημίου Πειραιά

Μέσω της μεθόδου `respond` ο χρήστης έχει τη δυνατότητα να αλλάζει τη γλώσσα σε όλες της οθόνες της εφαρμογής. Επιλέγοντας στο πάνω μέρος της οθόνης (toolbar) του τις τρεις τελείες (:) βλέπει 2 items. Με το πρώτο αλλάζει η γλώσσα σε όλη την εφαρμογή από ελληνικά σε αγγλικά και επιλέγοντας το δεύτερο, πηγαίνει στην οθόνη που αναφέρονται στοιχεία για την εφαρμογή. Παρακάτω φαίνονται οι σχετικές εικόνες.



Εικόνα 18. Homepage. – toolbar-

Όπως φαίνεται παρακάτω αλλάζουν τα ονόματα σε όλα τα UI στοιχεία καθώς επίσης και στα μηνύματα που βλέπει ο χρήστης σε όλη την εφαρμογή.



Εικόνα 19. Homepage. – Μεταβολή γλώσσας-

Εάν ο χρήστης έχει ήδη καταχωρήσει τα στοιχεία του (φύλο, ηλικία κτλ.) η Homepage που βλέπει είναι η παραπάνω. Αν από αυτή την οθόνη επιλέξει το button «Εμφάνιση Ανάλυσης», κατευθύνεται στην οθόνη της Fragment4, ενώ στην περίπτωση που επιλέξει το «Επανεκκινήστε την ανάλυση σώματος» διαγράφει τα data που έχει δώσει ο χρήστης και

κάνει refresh στη MainActivity, όπου εμφανίζονται πάλι τα buttons επιλογής φύλου (Fragment1).

Αν ο χρήστης επιλέξει το item «Σχετικά με μας», θα οδηγηθεί στο fragment (AboutFragment) στο οποίο παρουσιάζονται πληροφορίες σχετικά με την εφαρμογή.

5.2.4. Επιλογή Φύλου χρήστη - Fragment1

Όταν επιλεγθεί από την main activity στην οποία όταν μπαίνει ο χρήστης αρχικά, ο χρήστης επιλέγει το φύλο του (γυναίκα ή άνδρας) μέσω των 2 buttons που υπάρχουν στο Fragment1.



Εικόνα 20. Homepage - Fragment1.

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη `Fragment1` περιλαμβάνουν:

- «Buttons» (b1, b2). & «TextViews» αντικείμενα (tv1, tv2) για την επιλογή του φύλου και για την εμφάνιση κειμένου.
- «View» αντικείμενο (view) για την αναφορά στην ιεραρχία προβολής του fragment.
- Το «FragmentsCommunicator», για την επικοινωνία με άλλες δραστηριότητες ή fragments.

- Ένα «Typeface» αντικείμενο για την προσαρμογή της γραμματοσειράς των στοιχείων UI.

Μέθοδοι

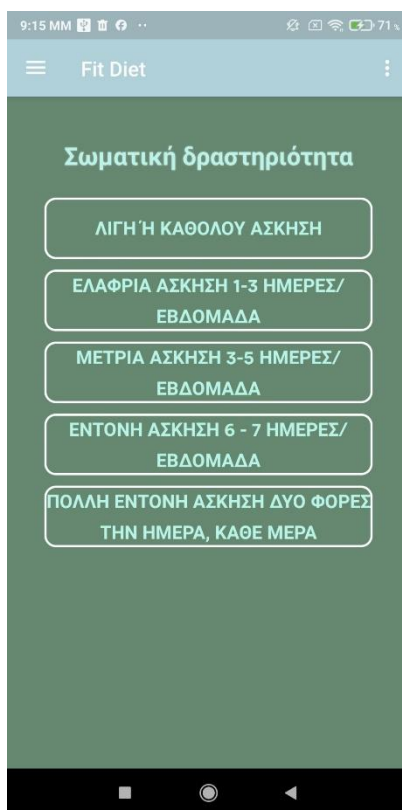
Η μέθοδος «onCreateView» καλείται όταν το fragment δημιουργείται. Στην «onCreateView», το layout του fragment αρχικοποιούνται τα στοιχεία UI όπως τα buttons και τα text views. Η μέθοδος αυτή φορτώνει μια προσαρμοσμένη γραμματοσειρά από τα assets και την εφαρμόζει στα text views και τα buttons. Επίσης, ορίζονται οι «OnClickListener» για τα buttons.

```
@Override
public void onClick(View view) {
    fc = (FragmentsCommunicator) getActivity();

    if (view.getId() == R.id.maleBtn){
        fc.respond("male", 0);
    }else if (view.getId() == R.id.femaleBtn){
        fc.respond("female", 0);
    }
}
```

5.2.5. Οθόνη Επιλογής επιπέδου φυσικής δραστηριότητας χρήστη - Fragment2

Στη συνέχεια, από τη Fragment1 στην οποία όταν ο χρήστης έχει ήδη επιλέξει το φύλο του πηγαίνει στο Fragment2 και επιλέγει το επίπεδο φυσικής τους δραστηριότητας μέσω πέντε κουμπιών. Περιλαμβάνει την αρχικοποίηση των στοιχείων UI και τη διαχείριση των κλικ των κουμπιών για την επικοινωνία με άλλες δραστηριότητες ή fragments. Ακολουθεί μια ανάλυση της λειτουργικότητας της κλάσης:



Εικόνα 21. Fragment2.

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη Fragment2 περιλαμβάνουν:

- Button για την επιλογή του επιπέδου φυσικής δραστηριότητας.
- Το FragmentsCommunicator για την επικοινωνία με άλλες δραστηριότητες ή fragments.
- Ένα View αντικείμενο (view) για την αναφορά στην ιεραρχία προβολής του fragment.
- Μια ακέραια μεταβλητή (phact) για την αποθήκευση της επιλεγμένης φυσικής δραστηριότητας.

Μέθοδοι

Η μέθοδος onCreateView καλείται όταν το fragment δημιουργείται. Στην onCreateView, το layout του fragment αρχικοποιούνται τα buttons. Η μέθοδος αυτή φορτώνει μια προσαρμοσμένη γραμματοσειρά από τα assets και την εφαρμόζει στα buttons. Επίσης, ορίζονται οι OnClickListener για τα buttons.

Η μέθοδος onClick χειρίζεται τα κλικ των κουμπιών. Ανάλογα με το κουμπί που πατήθηκε, καλεί τη μέθοδο respond του FragmentsCommunicator για να μεταδώσει τα δεδομένα στην δραστηριότητα ή σε άλλα fragments.

Java

```
@Override
public void onClick(View view) {
    fc = (FragmentsCommunicator) getActivity();
    if (view.getId() == R.id.button6) {
        fc.respond("SEDENTARY", 0);
    } else if (view.getId() == R.id.button7) {
        fc.respond("LIGHTLY ACTIVE", 0);
    } else if (view.getId() == R.id.button8) {
        fc.respond("MODERATELY ACTIVE", 0);
    } else if (view.getId() == R.id.button9) {
        fc.respond("VERY ACTIVE", 0);
    } else if (view.getId() == R.id.button10) {
        fc.respond("SUPER ACTIVE", 0);
    }
}
```

Η Fragment2 κλάση παρέχει μια απλή διεπαφή για την επιλογή του επιπέδου φυσικής δραστηριότητας του χρήστη και τη μετάδοση αυτής της επιλογής σε άλλα μέρη της εφαρμογής μέσω του FragmentsCommunicator. Οι προσαρμογές στη γραμματοσειρά και η διαχείριση των κλικ των κουμπιών εξασφαλίζουν μια ομοιόμορφη και λειτουργική εμπειρία χρήστη.

5.2.6. Οθόνη Επιλογής στόχου χρήστη- Fragment6

Η class Fragment6 επιτρέπει άλλες χρήστες να επιλέξουν τον στόχο άλλες, άλλες απώλεια βάρους, οικοδόμηση μυϊκής μάζας ή διατήρηση του τρέχοντος βάρους. Περιλαμβάνει την αρχικοποίηση των στοιχείων UI και τη διαχείριση των κλικ των κουμπιών για την επικοινωνία με άλλες δραστηριότητες ή fragments.

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη Fragment6 περιλαμβάνουν:

- Button για την επιλογή του στόχου (απώλεια βάρους, αύξηση μυϊκής μάζας, διατήρηση βάρους).
- View αντικείμενο (view) για την αναφορά στο layout του fragment.
- FragmentsCommunicator αντικείμενο για την επικοινωνία με άλλες δραστηριότητες ή fragments.
- Typeface για την εφαρμογή προσαρμοσμένης γραμματοσειράς στα buttons.
- String μεταβλητή (goal) για την αποθήκευση του επιλεγμένου στόχου.

Μέθοδοι

Η μέθοδος `onCreateView` καλείται όταν το `fragment` δημιουργείται. Στην `onCreateView`, το `layout` του `fragment` αρχικοποιούνται τα UI στοιχεία. Η μέθοδος αυτή φορτώνει τα `buttons` και τη γραμματοσειρά, και ορίζει τους `listeners` για τα `buttons`.

Η μέθοδος `onClick` χειρίζεται το κλικ των κουμπιών. Ανάλογα με το κουμπί που πατήθηκε, καλεί τη μέθοδο `respond` του `FragmentsCommunicator` για να μεταδώσει τον επιλεγμένο στόχο στην δραστηριότητα ή σε άλλα `fragments`.

Java

```
@Override
public void onClick(View view) {
    fc = (FragmentsCommunicator) getActivity();

    if (view.getId() == R.id.lose){
        fc.respond("lose", 0);
    } else if (view.getId() ==R.id.build ) {
        fc.respond("build", 0);
    } else if (view.getId() ==R.id.maintain){
        fc.respond("maintain", 0);
    }
}
}
```

Η κλάση `Fragment6` παρέχει μια απλή διεπαφή για την επιλογή στόχων του χρήστη και τη μετάδοση αυτής της πληροφορίας σε άλλα μέρη της εφαρμογής μέσω του `FragmentsCommunicator`. Η διαχείριση των κλικ των κουμπιών εξασφαλίζει ότι ο χρήστης μπορεί εύκολα να επιλέξει τον στόχο του και να προχωρήσει στην επόμενη ενέργεια στην εφαρμογή.



Εικόνα 22. Fragment6.

5.2.7. Οθόνη για Καταχώρηση ηλικίας, ύψους και βάρους χρήστη - Fragment3

Η κλάση Fragment3 επιτρέπει στους χρήστες να εισάγουν την ηλικία, το βάρος και το ύψος τους. Περιλαμβάνει την αρχικοποίηση των στοιχείων UI και τη διαχείριση των κλικ των κουμπιών για την επικοινωνία με άλλες δραστηριότητες ή fragments. Ακολουθεί μια ανάλυση της λειτουργικότητας της κλάσης:

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη Fragment3 περιλαμβάνουν:

- Button για την επιβεβαίωση των εισαγόμενων δεδομένων, με το οποίο μεταβαίνει στη Fragment4 .
- Ένα NumberPicker αντικείμενο (noPicker) για την επιλογή της ηλικίας.
- Ένα FragmentsCommunicator αντικείμενο για την επικοινωνία με άλλες δραστηριότητες ή fragments.
- Δύο EditText αντικείμενα για την εισαγωγή του βάρους και του ύψους.
- Μια ακέραια μεταβλητή (age) για την αποθήκευση της επιλεγμένης ηλικίας.

Μέθοδοι

Η μέθοδος onCreateView καλείται όταν το fragment δημιουργείται. Στην onCreateView, το layout του fragment αρχικοποιούνται τα UI στοιχεία. Η μέθοδος αυτή φορτώνει το FragmentsCommunicator, τα buttons, τα πεδία κειμένου και το NumberPicker.

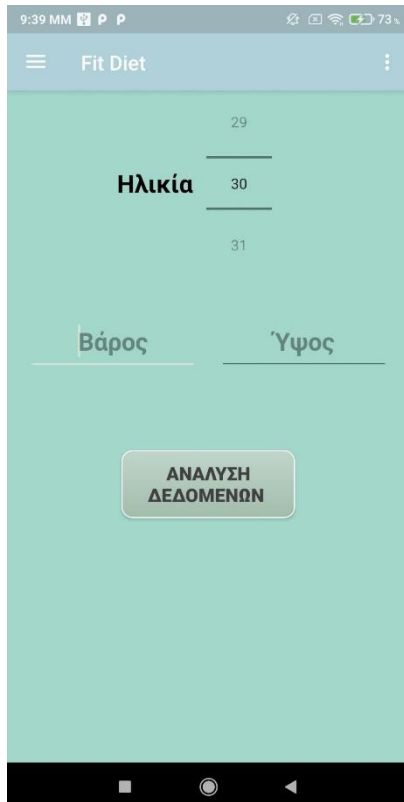
Η μέθοδος onClick χειρίζεται το κλικ του κουμπιού. Ανάλογα με τα δεδομένα που εισάγονται από τον χρήστη, ελέγχει αν έχουν συμπληρωθεί σωστά όλα τα πεδία και αν είναι έγκυρα. Στη συνέχεια, καλεί τη μέθοδο respond του FragmentsCommunicator για να μεταδώσει τα δεδομένα στην δραστηριότητα ή σε άλλα fragments.

Java

```
@Override
public void onClick(View view) {
    fc = (FragmentsCommunicator) getActivity();
    if (age == 0){
        Toast.makeText(getActivity(),
getString(R.string.enterAge), Toast.LENGTH_SHORT).show();
    }else if (et1.getText().toString().matches("")) {
        Toast.makeText(getActivity(),
getString(R.string.enterweight), Toast.LENGTH_SHORT).show();
    }else if (Integer.valueOf(et1.getText().toString()) < 20 ||
Integer.valueOf(et1.getText().toString()) > 250){
        Toast.makeText(getActivity(),
getString(R.string.enterWeight2), Toast.LENGTH_SHORT).show();
    }else if (et2.getText().toString().matches("")){
```

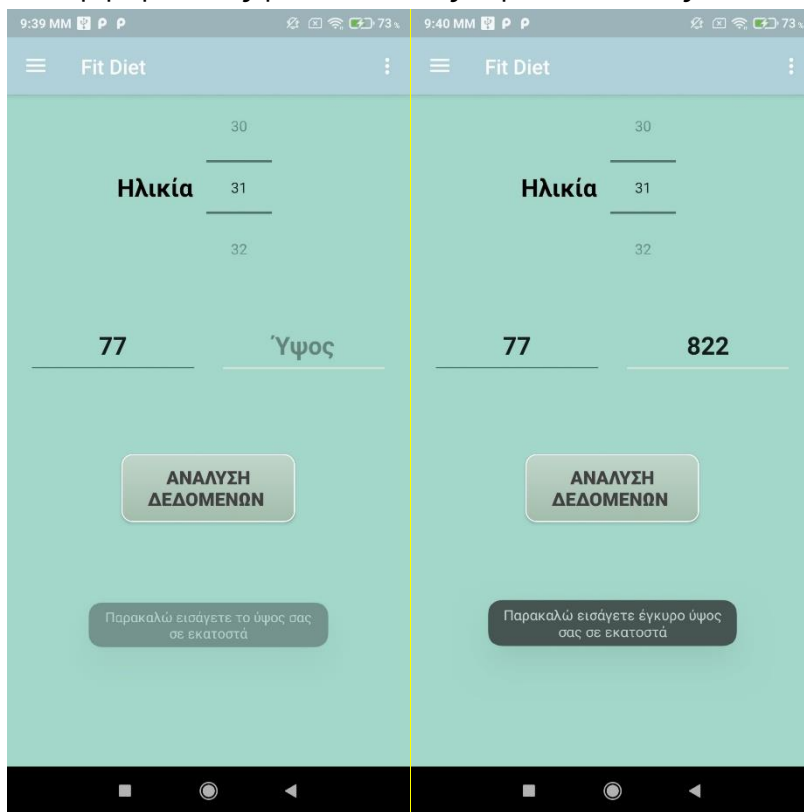
```
        Toast.makeText(getActivity(),  
getString(R.string.enterHeight), Toast.LENGTH_SHORT).show();  
    }else if(Integer.valueOf(et2.getText().toString()) < 60 ||  
Integer.valueOf(et2.getText().toString()) > 220){  
        Toast.makeText(getActivity(),  
getString(R.string.enterHeight2), Toast.LENGTH_SHORT).show();  
    }  
    else{  
        fc.respond("age", age );  
        fc.respond("weight",  
Integer.valueOf(et1.getText().toString()));  
        fc.respond("height",  
Integer.valueOf(et2.getText().toString()));  
        fc.respond("analyze", 0);  
    }  
}
```

Η κλάση Fragment3 παρέχει μια απλή διεπαφή για την εισαγωγή βασικών δεδομένων υγείας του χρήστη και τη μετάδοση αυτής της πληροφορίας σε άλλα μέρη της εφαρμογής μέσω του FragmentsCommunicator. Οι έλεγχοι εγκυρότητας και η διαχείριση των κλικ των κουμπιών εξασφαλίζουν ότι τα δεδομένα του χρήστη είναι σωστά και έγκυρα πριν από την αποστολή τους.

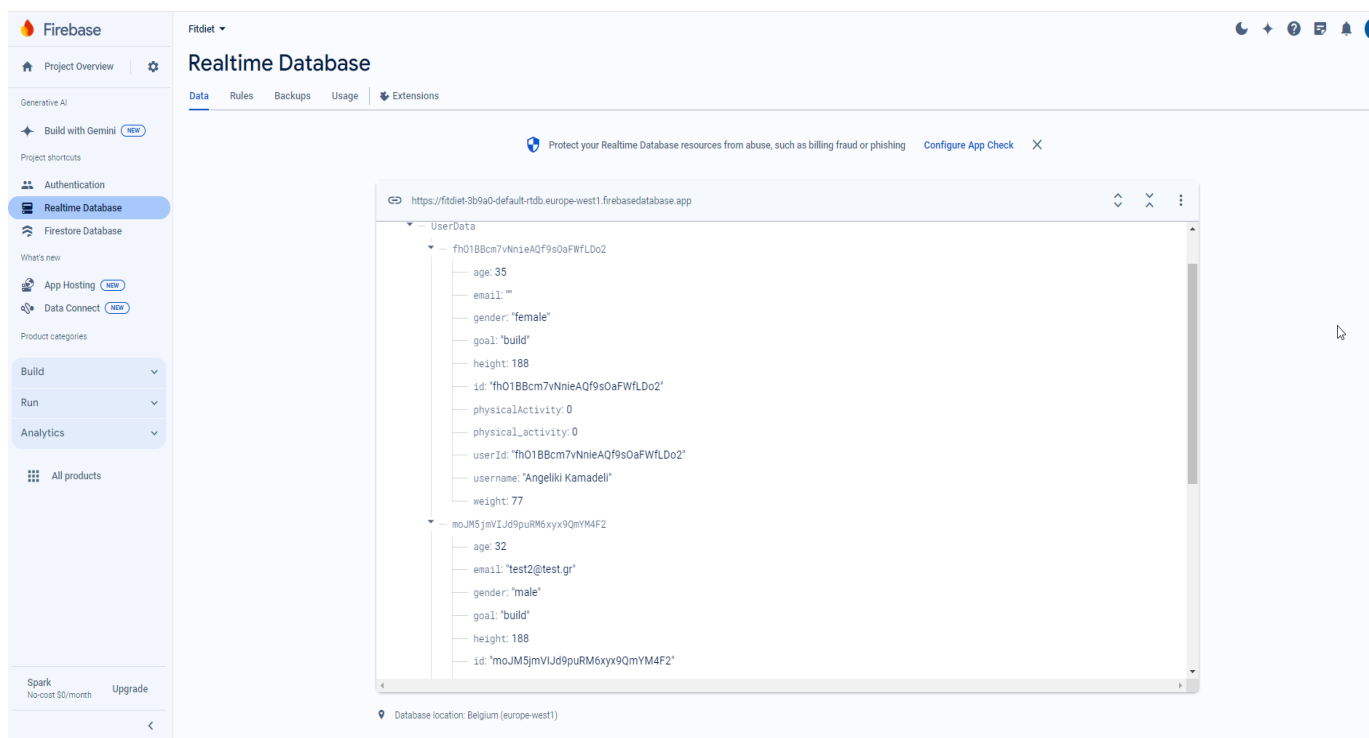


Εικόνα 23. Fragment3.

Σε περίπτωση που καταχωρήσει λάθος μορφή για το ύψος ή δεν καταχωρήσει τιμή, βγαίνει ενημερωτικό μήνυμα όπως φαίνεται και στις παρακάτω εικόνες.

**Εικόνες 24-25. Fragment3.**

Τα στοιχεία που έχει καταχωρήσει ο χρήστης στα παραπάνω Fragments (Fragment1, Fragment2, Fragment6, Fragment3) καταχωρούνται στη Realtime Firebase με την χρήση της class FirebaseManager. Στην εικόνα παρακάτω, φαίνονται τα στοιχεία των πινάκων που έχουν δημιουργηθεί στη Firebase.



Εικόνα 26. Realtime Database. UserData

5.2.8. Οθόνη Υπολογισμός και εμφάνιση ΔΜΣ, Ημερήσια κατανάλωση νερού και θερμίδων - Fragment4

Στην κλάση Fragment4, το SharedPreferences χρησιμοποιείται για την αποθήκευση και ανάκτηση δεδομένων χρήστη, όπως η ηλικία, το βάρος και το ύψος. Αυτά τα δεδομένα είναι απαραίτητα για διάφορους υπολογισμούς που γίνονται στην εφαρμογή και πρέπει να διατηρούνται ακόμα και όταν η εφαρμογή κλείνει.

Στην αρχή του onCreateView, αρχικοποιούνται τα διάφορα UI components, όπως κουμπιά (buttons), TextViews και SeekBar. Αυτά τα components χρησιμοποιούνται για την προβολή των δεδομένων και την αλληλεπίδραση με τον χρήστη. Για παράδειγμα, υπάρχουν κουμπιά για την εμφάνιση της ποσότητας νερού που χρειάζεται ο χρήστης ημερησίως και των θερμίδων που πρέπει να καταναλώνει.

Τα δεδομένα σχετικά με την ηλικία, το βάρος, το ύψος, τη φυσική δραστηριότητα, τον στόχο και το φύλο ανακτώνται από την MainActivity. Αυτά τα δεδομένα είναι απαραίτητα για τους υπολογισμούς που θα γίνουν αργότερα.

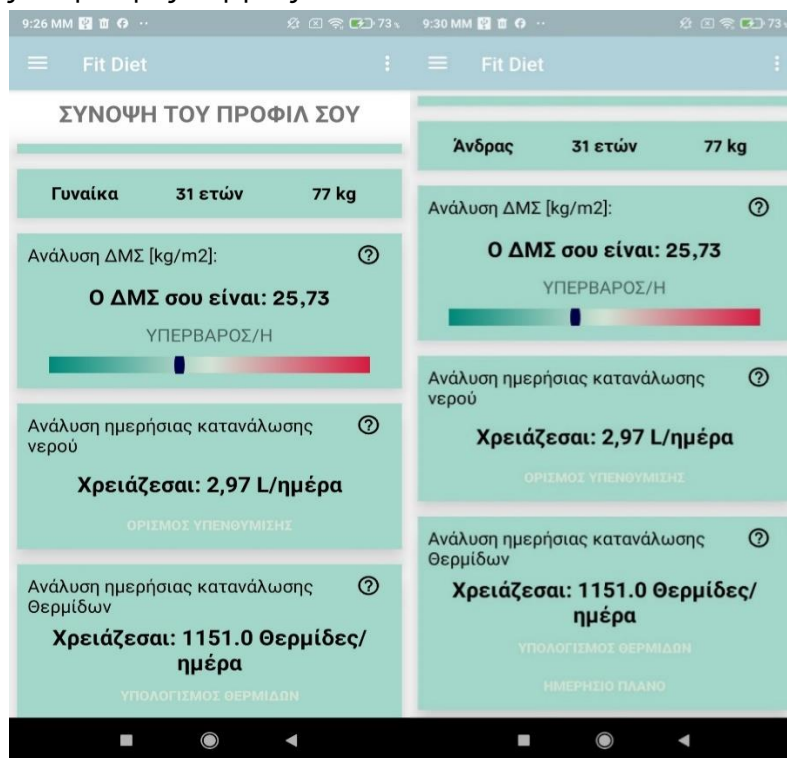
Ο Δείκτης Μάζας Σώματος (BMI) υπολογίζεται και εμφανίζεται στο αντίστοιχο TextView. Ο υπολογισμός του BMI γίνεται με βάση το βάρος και το ύψος του χρήστη. Αναλόγως της τιμής του BMI, εμφανίζεται και μια σχετική ετικέτα, όπως "παχυσαρκία" ή "φυσιολογικό βάρος".

Στη συνέχεια, υπολογίζεται η απαιτούμενη ημερήσια πρόσληψη νερού με βάση την ηλικία και το βάρος του χρήστη. Το αποτέλεσμα αυτών των υπολογισμών εμφανίζεται στην οθόνη, βοηθώντας τον χρήστη να κατανοήσει πόσο νερό πρέπει να καταναλώνει καθημερινά.


Οι ημερήσιες θερμίδες που χρειάζεται ο χρήστης υπολογίζονται με βάση τη φυσική του δραστηριότητα, τον στόχο του (π.χ. απώλεια βάρους, διατήρηση βάρους, αύξηση μυϊκής

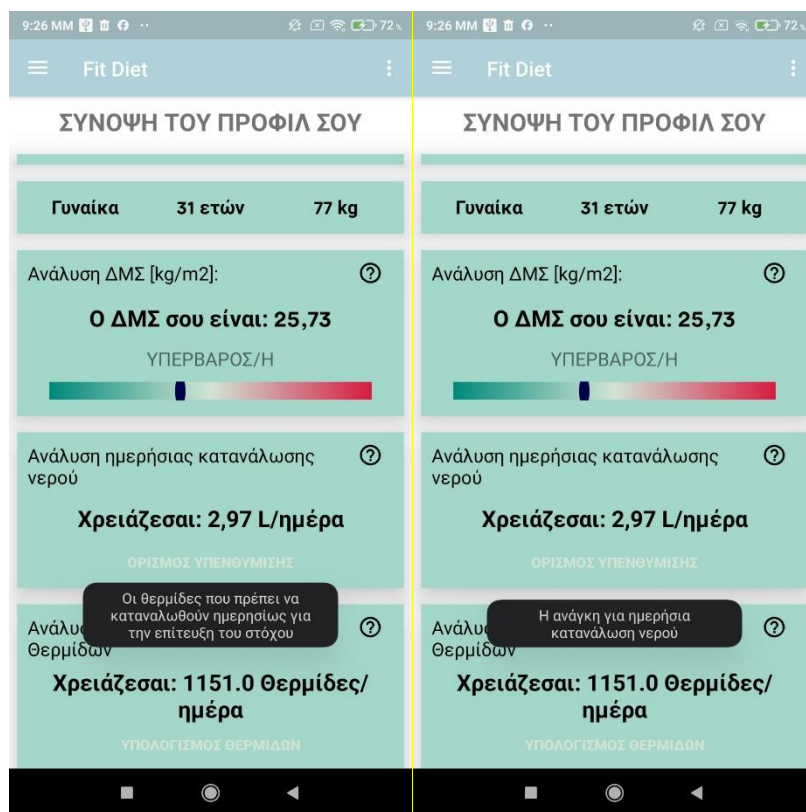
μάζας) και το φύλο του. Οι θερμίδες αυτές εμφανίζονται στην οθόνη και προσαρμόζονται αναλόγως των παραμέτρων που έχουν οριστεί από τον χρήστη.

Τέλος, τα διάφορα κουμπιά και τα ImageView έχουν OnClickListener, οι οποίοι διαχειρίζονται τα κλικ των χρηστών. Με αυτόν τον τρόπο, όταν ο χρήστης πατάει κάποιο κουμπί, η εφαρμογή ανταποκρίνεται κατάλληλα, προβάλλοντας τα απαιτούμενα δεδομένα ή εκτελώντας τις απαραίτητες ενέργειες.



Εικόνες 27-28. Fragment4.

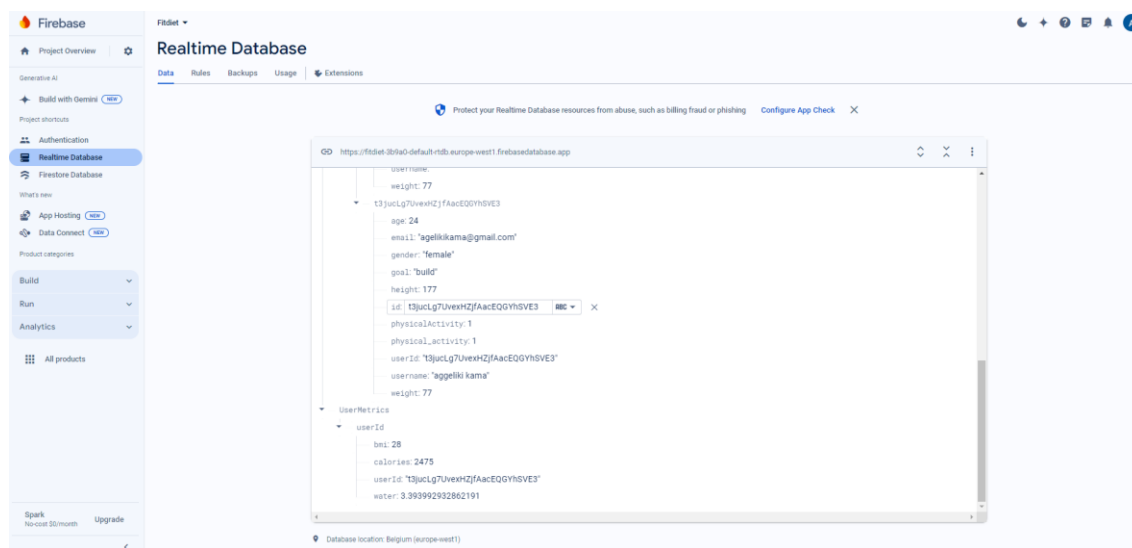
Όπως φαίνεται και στην εικόνα παραπάνω, κάτω από την ανάλυση του ΔΜΣ, εμφανίζεται η ετικέτα και κάτω από την ετικέτα υπάρχει ένα SeekBar στο οποίο υπάρχει οπτική απεικόνιση για το ΔΜΣ. Εάν ο χρήστης θέλει περισσότερες πληροφορίες σχετικά με το κάθε Fragment που βλέπει (ήτοι το Fragment με τα στοιχεία για το ΔΜΣ, το Fragment με τα στοιχεία για την ημερήσια κατανάλωση νερού και το Fragment με τα στοιχεία για την ημερήσια κατανάλωση θερμίδων) υπάρχει ένα ImageView  που αν το πατήσει ο χρήστης εμφανίζεται ενημερωτικό μήνυμα όπως φαίνεται στις παρακάτω εικόνες.



Εικόνες 29-30. Fragment4 – Ενημερωτικά μηνύματα.

Αν ο χρήστης επιλέξει ένα από τα buttons που αφορούν την ημερήσια κατανάλωση νερού (waterBtn) και θερμίδων (caloriesBtn & buttonShowMealPlan) αντίστοιχα θα οδηγηθεί στην αντίστοιχη οθόνη. Για παράδειγμα, αν επιλέξει το button «Ορισμός Υπενθύμισης», θα οδηγηθεί στην οθόνη της RemindersActivity, όπου μπορεί να ορίσει κάθε πόση ώρα θα έχει υπενθύμιση για να καταναλώσει την απαιτούμενη ποσότητα νερού ώστε να επιτευχθεί ο ημερήσιος στόχος για καταναλωθεί η αναγκαία ποσότητα νερού. Αντίστοιχα, αν επιλέξει τον «Υπολογισμό θερμίδων», θα οδηγηθεί στην οθόνη της ApiActivity ενώ αν επιλέξει το «Ημερήσιο Πλάνο» θα οδηγηθεί στην οθόνη της MealPlanFragment.

Στην παρακάτω εικόνα φαίνονται τα στοιχεία που έχουν καταχωρηθεί στη Realtime Firebase για το χρήστη στον UserMetrics.



Εικόνα 31. Firebase - table UserMetrics.

5.2.9. FirebaseManager

Στην κλάση `FirebaseManager`, υπάρχουν σταθερές και μεταβλητές που χρησιμοποιούνται για τη διαχείριση της σύνδεσης με τη Firebase και την αποθήκευση δεδομένων χρηστών.

Σταθερές και Μεταβλητές

Οι βασικές μεταβλητές είναι οι `database`, `userRef` και `metricsRef`. Η μεταβλητή `database` είναι μια αναφορά στη βάση δεδομένων της Firebase, η οποία αρχικοποιείται με το URL της βάσης δεδομένων. Οι `userRef` και `metricsRef` είναι αναφορές στα τμήματα της βάσης δεδομένων όπου αποθηκεύονται τα δεδομένα χρηστών και τα μετρητικά στοιχεία τους, αντίστοιχα.

Μέθοδοι

Η μέθοδος `FirebaseManager` είναι ο κατασκευαστής της κλάσης και αρχικοποιεί τις βασικές μεταβλητές που αναφέρθηκαν παραπάνω. Εντός αυτής της μεθόδου, γίνεται σύνδεση με τη Firebase και ορίζονται οι διαδρομές για τα δεδομένα χρηστών και τα μετρητικά στοιχεία.

Η μέθοδος `saveUserData` αποθηκεύει τα δεδομένα χρήστη στη βάση δεδομένων της Firebase. Λαμβάνει ως παραμέτρους το `userId`, το `username`, το `email`, την ηλικία, το φύλο, το ύψος, το βάρος, τη φυσική δραστηριότητα, τον στόχο και τον τύπο σύνδεσης. Δημιουργεί ένα πίνακα `UserData` και το αποθηκεύει στη βάση δεδομένων στη διαδρομή που ορίζεται από το `userId`.

Η μέθοδος `calculateAndSaveUserMetrics` υπολογίζει και αποθηκεύει τα μετρητικά στοιχεία χρήστη, όπως το BMI, την απαιτούμενη ποσότητα νερού και τις θερμίδες που χρειάζεται ο χρήστης. Οι υπολογισμοί γίνονται με βάση την ηλικία, το ύψος, το βάρος, το φύλο, τη φυσική δραστηριότητα και τον στόχο του χρήστη. Μετά τους υπολογισμούς, δημιουργείται ένας

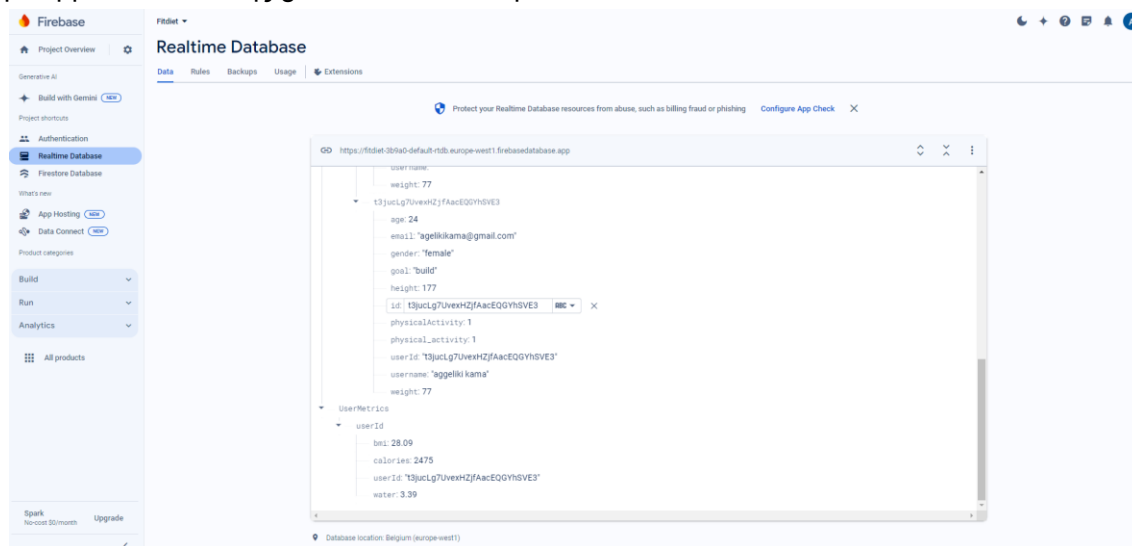
πίνακας `UserMetrics` και αποθηκεύεται στη βάση δεδομένων στη διαδρομή που ορίζεται από το `userId`.

Η μέθοδος `getUserData` ανακτά τα δεδομένα χρήστη από τη Firebase βάση δεδομένων. Λαμβάνει ως παράμετρο το `userId` και ένα αντικείμενο `FirestoreCallback` για να διαχειριστεί την επιστροφή των δεδομένων. Χρησιμοποιεί έναν `ValueEventListener` για να ακούσει τις αλλαγές στα δεδομένα και να καλέσει την κατάλληλη μέθοδο του `callback` όταν τα δεδομένα είναι διαθέσιμα ή αν υπάρξει κάποιο σφάλμα.

Η μέθοδος `getUserMetrics` λειτουργεί παρόμοια με την `getUserData`, αλλά ανακτά τα μετρητικά στοιχεία του χρήστη. Λαμβάνει επίσης ως παράμετρο το `userId` και ένα αντικείμενο `FirestoreCallback`. Χρησιμοποιεί έναν `ValueEventListener` για να ακούσει τις αλλαγές στα δεδομένα μετρήσεων και να καλέσει την κατάλληλη μέθοδο του `callback` όταν τα δεδομένα είναι διαθέσιμα ή αν υπάρξει κάποιο σφάλμα.

Η εσωτερική διεπαφή `FirestoreCallback` ορίζεται για να χειρίζεται την επιστροφή δεδομένων από τις μεθόδους ανάκτησης δεδομένων. Περιλαμβάνει δύο μεθόδους: `onCallback` για να διαχειριστεί την επιτυχή επιστροφή δεδομένων και `onError` για να διαχειριστεί τυχόν σφάλματα που προκύπτουν κατά την ανάκτηση δεδομένων.

Τέλος, στην κλάση `FirestoreManager` χρησιμοποιούνται οι `classes` `UserData` και `UserMetrics`. Η κλάση `UserData` περιλαμβάνει τα δεδομένα χρήστη όπως το `id`, το `username`, το `email`, την ηλικία, το φύλο, το ύψος, το βάρος, τη φυσική δραστηριότητα, τον στόχο και τον τύπο σύνδεσης. Ομοίως, στην κλάση `UserMetrics` περιλαμβάνονται τα μετρητικά στοιχεία χρήστη όπως το `BMI`, την απαιτούμενη ποσότητα νερού και τις θερμίδες, καθώς και το `userId` του χρήστη. Και στις 2 `classes` υπάρχουν δύο κατασκευαστές, ένας χωρίς παραμέτρους και ένας με παραμέτρους για την αρχικοποίηση των δεδομένων χρήστη. Περιλαμβάνονται επίσης `getters` και `setters` για κάθε πεδίο.



Εικόνα 32. Firebase UserData & UserMetrics

5.2.10. Οθόνη Υπενθύμισης κατανάλωσης νερού - RemindersActivity

Η class «RemindersActivity» είναι υπεύθυνη για τη διαχείριση των υπενθυμίσεων στην εφαρμογή, επιτρέποντας στους χρήστες να ρυθμίσουν χρονικά διαστήματα υπενθυμίσεων και να ξεκινήσουν ή να σταματήσουν τις υπενθυμίσεις. Ακολουθεί μια ανάλυση της λειτουργικότητας της κλάσης:

Σταθερές και Μεταβλητές

Οι σταθερές και μεταβλητές που χρησιμοποιούνται στη `RemindersActivity` περιλαμβάνουν:

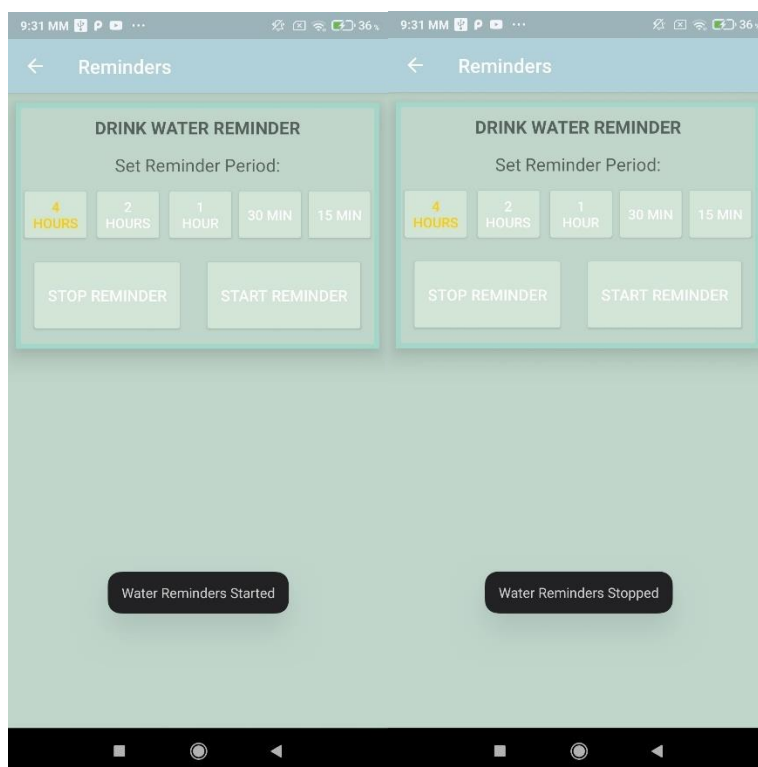
- Διάφορα UI Buttons που αντιπροσωπεύουν τα buttons για την επιλογή χρονικών διαστημάτων και τον έλεγχο των υπενθυμίσεων.
- Ένα `SharedPreferences` αντικείμενο για την αποθήκευση των ρυθμίσεων υπενθυμίσεων του χρήστη.

Όπως φαίνεται και στην κάτωθι οθόνη, ο χρήστης επιλέγει κάθε πόση ώρα θα έχει υπενθύμιση για κατανάλωση νερού



Εικόνα 33. Οθόνη Υπενθυμίσεων - RemindersActivity

Στις παρακάτω οθόνες φαίνονται τα μηνύματα που βλέπει ο χρήστης όταν ξεκινάει την υπενθύμιση για την κατανάλωση νερού και αντίστοιχα σταματάει το reminder με το ενημερωτικό μήνυμα.



Εικόνες 34-35. Οθόνη Υπενθυμίσεων -Start Reminder & Stop Reminder

Μέθοδοι

Η μέθοδος `onCreate` καλείται όταν η δραστηριότητα δημιουργείται. Στην `onCreate`, αρχικοποιούνται τα στοιχεία UI και το `SharedPreferences`, και ορίζονται οι `OnClickListener` για τα buttons. Η `onCreate` μέθοδος περιλαμβάνει επίσης την αρχικοποίηση του `ActionBar` και την ορισμό του τίτλου του.

Η μέθοδος `onClick` χειρίζεται τα κλικ των κουμπιών. Ανάλογα με το κουμπί που πατήθηκε, ρυθμίζει την καθυστέρηση υπενθύμισης και αλλάζει το χρώμα του επιλεγμένου κουμπιού. Οι `startReminder` και `stopReminder` μέθοδοι καλούνται επίσης για να ξεκινήσουν ή να σταματήσουν τις υπενθυμίσεις αντίστοιχα.

Η μέθοδος `updateButtonColors` αλλάζει το χρώμα του κειμένου του επιλεγμένου κουμπιού σε κίτρινο και τα υπόλοιπα σε λευκό.

Java

```
private void updateButtonColors(Button selectedButton) {
    b1.setTextColor(b1 == selectedButton ?
    Color.parseColor("#FFCC00") : Color.WHITE);
    b2.setTextColor(b2 == selectedButton ?
    Color.parseColor("#FFCC00") : Color.WHITE);
    b3.setTextColor(b3 == selectedButton ?
    Color.parseColor("#FFCC00") : Color.WHITE);
    b4.setTextColor(b4 == selectedButton ?
    Color.parseColor("#FFCC00") : Color.WHITE);
    b5.setTextColor(b5 == selectedButton ?
```

```
Color.parseColor("#FFCC00") : Color.WHITE);
}
```

Η μέθοδος `startReminder` ξεκινάει την υπενθύμιση χρησιμοποιώντας `AlarmManager` για την ενεργοποίηση ενός `PendingIntent` σε καθορισμένα διαστήματα. Αντίστοιχα Η μέθοδος `stopReminder` σταματάει την υπενθύμιση ακυρώνοντας το `PendingIntent`.

Java

```
private void startReminder(int delay) {
    AlarmManager alarmManager = (AlarmManager)
    getSystemService(Context.ALARM_SERVICE);
    Intent alarmIntent = new Intent(this, AlarmReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(this,
    0, alarmIntent, PendingIntent.FLAG_UPDATE_CURRENT |
    PendingIntent.FLAG_IMMUTABLE);
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,
    System.currentTimeMillis(), delay * 60 * 1000, pendingIntent);
    Toast.makeText(this, "Water Reminders Started",
    Toast.LENGTH_SHORT).show();
}
```

Αυτή η κλάση παρέχει μια ολοκληρωμένη διαχείριση υπενθυμίσεων, επιτρέποντας στους χρήστες να επιλέξουν το διάστημα υπενθύμισης και να ελέγξουν την ενεργοποίηση ή απενεργοποίηση αυτών των υπενθυμίσεων. Χρησιμοποιεί την class AlarmReceiver για την ειδοποίηση που θα δώσει.

5.2.11. Οθόνη υπολογισμού θερμίδων - ApiActivity

Η κλάση ApiActivity είναι μια υποκλάση της AppCompatActivity και διαχειρίζεται την αλληλεπίδραση με μια εξωτερική API για την ανάκτηση διατροφικών πληροφοριών, καθώς και την αποθήκευση και διαχείριση ημερήσιων θερμίδων του χρήστη. Χρησιμοποιεί διάφορες μεταβλητές για την αποθήκευση δεδομένων και την αναφορά σε διαφορετικά στοιχεία του UI και τη Firebase βάση δεδομένων.

Σταθερές και Μεταβλητές

Οι σταθερές και οι μεταβλητές που χρησιμοποιούνται στην κλάση περιλαμβάνουν:

- Prefs_NAME και CALORIE_COUNTER_KEY, που είναι σταθερές για την αποθήκευση των θερμίδων στις SharedPreferences.
- sharedPreferences, η οποία χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων θερμίδων.
- food, μια μεταβλητή που αποθηκεύει το όνομα του φαγητού που εισάγει ο χρήστης.
- et, tv1, tv2, tv3, tv4, tv5, tv6, v1, και v2, οι οποίες είναι αναφορές στα στοιχεία του UI.
- calories, μια μεταβλητή που αποθηκεύει τις θερμίδες του τρέχοντος φαγητού.

- `dailyCaloriesRef`, μια αναφορά στη Firebase βάση δεδομένων για την αποθήκευση ημερήσιων θερμίδων.

Η κλάση επίσης περιλαμβάνει τις μεταβλητές `BASE_URL`, `retrofit`, και `calorieNinjasApi`, οι οποίες χρησιμοποιούνται για την επικοινωνία με την εξωτερική API `CalorieNinjas`.

Μέθοδοι

Στην μέθοδο `onCreate`, αρχικοποιούνται οι μεταβλητές και οι αναφορές στα στοιχεία του UI, καθώς και το `Retrofit` για την επικοινωνία με την API. Η μέθοδος `setContentView` φορτώνει το layout της δραστηριότητας και οι αναφορές στα στοιχεία του UI συνδέονται με τα αντίστοιχα στοιχεία του layout.

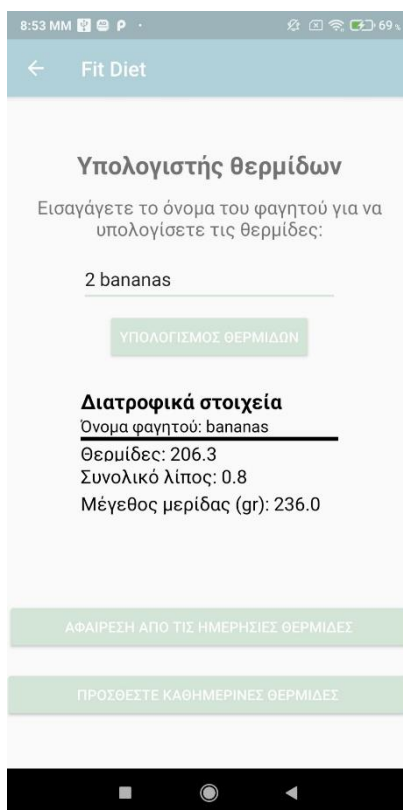
Η μέθοδος `calculateCk` ενεργοποιείται όταν ο χρήστης πατά το κουμπί για να υπολογίσει τις θερμίδες. Παίρνει το όνομα του φαγητού από το `EditText` και καλεί την API για να λάβει τις διατροφικές πληροφορίες. Αν η απάντηση είναι επιτυχής, ενημερώνονται τα `TextViews` με τις διατροφικές πληροφορίες του φαγητού και αποθηκεύονται οι θερμίδες στη μεταβλητή `calories`. Αν η κλήση στην API αποτύχει, εμφανίζεται ένα μήνυμα σφάλματος.

Οι μέθοδοι `addDailyCalories` και `removeDailyCalories` διαχειρίζονται την προσθήκη και την αφαίρεση των θερμίδων στην αποθήκευση των `SharedPreferences`. Οι μέθοδοι αυτές ανακτούν την τρέχουσα τιμή των θερμίδων από τις `SharedPreferences`, την ενημερώνουν με βάση τις θερμίδες του τρέχοντος φαγητού και αποθηκεύουν την ενημερωμένη τιμή πίσω στις `SharedPreferences`. Επίσης εμφανίζουν ένα `Toast` μήνυμα για να ενημερώσουν τον χρήστη για την επιτυχή προσθήκη ή αφαίρεση θερμίδων.

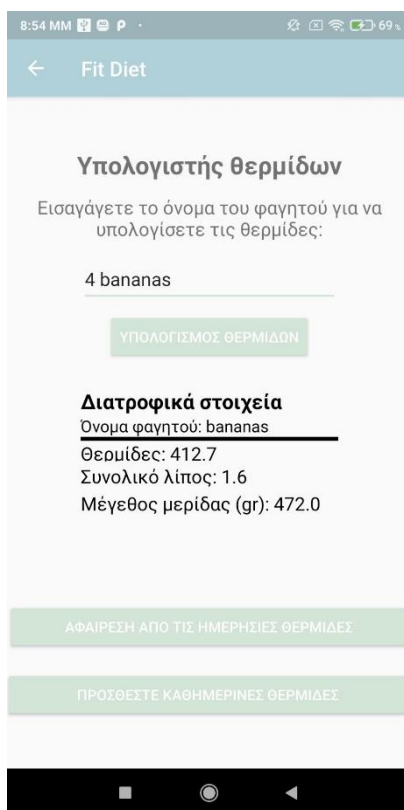
Το interface `CalorieNinjasApi` ορίζει την μέθοδο `getNutritionInfo`, η οποία χρησιμοποιεί το `Retrofit` για να καλέσει την API και να λάβει τις διατροφικές πληροφορίες. Αυτή η μέθοδος λαμβάνει ως παραμέτρους το κλειδί API και το όνομα του φαγητού που εισάγει ο χρήστης.

Επιπλέον, η `ApiActivity` χρησιμοποιεί την κλάση `NutritionResponse` για να αποθηκεύσει μια λίστα διατροφικών στοιχείων για το φαγητό που αναζητά ο χρήστης στην `ApiActivity`. Η λίστα της class `NutritionResponse` περιλαμβάνει, το όνομα του φαγητού που αναζητά ο χρήστης, τις θερμίδες, το συνολικό ποσό λίπους του φαγητού σε γραμμάρια και το μέγεθος του σε γραμμάρια. Στην κλάση `NutritionItem` υπάρχουν οι μέθοδοι `get` και `set` που αφορούν την παραπάνω λίστα διατροφικών στοιχείων για το φαγητό που αναζητά ο χρήστης. Οι μέθοδοι που χρησιμοποιούνται επιτρέπουν την ανάγνωση και εγγραφή των τιμών των μεταβλητών αυτών.

Όπως φαίνεται στις παρακάτω εικόνες, εάν ο χρήστης επιλέξει στην αναζήτησή του π.χ. 2 μπανάνες. Θα δει τα σχετικά διατροφικά στοιχεία, και σε περίπτωση που αυξήσει την ποσότητα τα στοιχεία θα ενημερωθούν ορθά.



Εικόνα 36. Οθόνη Αναζήτησης Διατροφικών στοιχείων - AriActivity



Εικόνες 37. Οθόνη Αναζήτησης Διατροφικών στοιχείων - ApiActivity

5.2.12. Οθόνη Συνταγών -Recipes

Η κλάση Recipes εφαρμόζει το interface FragmentsCommunicator. Αυτή η κλάση διαχειρίζεται την εμφάνιση και την αλληλεπίδραση με μια λίστα συνταγών, επιτρέποντας την εμφάνιση των συνταγών σε ένα RecyclerView και την πλοήγηση στις λεπτομέρειες των συνταγών.

Σταθερές και Μεταβλητές

Οι σταθερές και οι μεταβλητές που χρησιμοποιούνται στην κλάση περιλαμβάνουν:

Η μεταβλητή `tf` αποθηκεύει έναν τύπο γραμματοσειράς (Typeface) που φορτώνεται από τα assets της εφαρμογής. Η `recyclerView` είναι μια αναφορά στο στοιχείο RecyclerView του UI που εμφανίζει τις συνταγές, ενώ η `recipeAdapter` είναι μια μεταβλητή που αναφέρεται στον προσαρμογέα (adapter) που χρησιμοποιείται για τη διαχείριση των δεδομένων που θα εμφανίζονται στο RecyclerView. Η `recipeList` είναι μια λίστα (`List<Recipe>`) που αποθηκεύει τα δεδομένα των συνταγών.

Μέθοδοι

Η μέθοδος `newInstance` δημιουργεί και επιστρέφει μια νέα περίπτωση του `Recipes` fragment, αποθηκεύοντας μια ετικέτα (`tag`) στα `arguments` του fragment για μελλοντική χρήση.

Στη μέθοδο `onCreateView`, αρχικοποιείται ο τύπος γραμματοσειράς, γίνεται `inflate` το layout του fragment και δημιουργείται το `RecyclerView` που συνδέεται με τον αντίστοιχο προσαρμογέα (`adapter`). Η λίστα συνταγών (`recipeList`) αρχικοποιείται και η `recipeAdapter` ορίζεται να χρησιμοποιεί αυτή τη λίστα, καθώς και να αναφέρεται στη διεπαφή `FragmentsCommunicator` που εφαρμόζει το ίδιο το fragment. Τέλος, καλείται η μέθοδος `fetchRecipes` για την ανάκτηση των συνταγών με βάση την ετικέτα (`tag`) που αποθηκεύεται στα `arguments` του fragment.

Η μέθοδος `fetchRecipes` χρησιμοποιεί το `Retrofit` για να καλέσει μια εξωτερική API και να λάβει τις συνταγές. Η κλήση στην API γίνεται με βάση την ετικέτα (`tag`) που δίνεται ως παράμετρος στη μέθοδο. Εάν η απάντηση είναι επιτυχής και περιέχει δεδομένα, τα δεδομένα προστίθενται στη λίστα των συνταγών (`recipeList`) και ενημερώνεται ο προσαρμογέας (`adapter`) για να εμφανίσει τις νέες συνταγές στο `RecyclerView`. Σε περίπτωση αποτυχίας της κλήσης στην API, εμφανίζεται ένα μήνυμα σφάλματος στον χρήστη και καταγράφεται το σφάλμα στο `log`.

Η μέθοδος `goToRecipeDetails` υλοποιεί τη διεπαφή `FragmentsCommunicator` και χρησιμοποιείται για την πλοήγηση στις λεπτομέρειες μιας συνταγής. Δημιουργείται ένα νέο instance του `RecipeDetailsFragment` με τη συγκεκριμένη συνταγή και γίνεται αντικατάσταση του τρέχοντος fragment με το νέο fragment, προσθέτοντας την ενέργεια στο `back stack` ώστε ο χρήστης να μπορεί να επιστρέψει στην προηγούμενη οθόνη.

Η κλάση `RecipeAdapter` είναι ένας προσαρμογέας (`adapter`) για το `RecyclerView`, ο οποίος διαχειρίζεται τα δεδομένα των συνταγών και τις αλληλεπιδράσεις του χρήστη. Περιλαμβάνει μια λίστα συνταγών (`recipes`) και μια αναφορά στη διεπαφή `FragmentsCommunicator` για την επικοινωνία μεταξύ του `adapter` και του fragment. Στις μεθόδους `onCreateViewHolder` και `onBindViewHolder`, δημιουργούνται και ενημερώνονται οι θήκες προβολής (`view holders`) για κάθε συνταγή. Η μέθοδος `getItemCount` επιστρέφει τον αριθμό των συνταγών στη λίστα.

Η εσωτερική κλάση `RecipeViewHolder` είναι μια θήκη προβολής (`view holder`) που διαχειρίζεται τα στοιχεία του UI για κάθε συνταγή. Περιλαμβάνει αναφορές σε διάφορα στοιχεία του UI, όπως τα `TextView` για τον τίτλο της συνταγής, τον χρόνο μαγειρέματος, το σκορ υγείας και τις μερίδες, καθώς και μια εικόνα της συνταγής (`ImageView`). Η μέθοδος `bind` χρησιμοποιείται για την ενημέρωση αυτών των στοιχείων με τα δεδομένα της συνταγής και τη φόρτωση της εικόνας χρησιμοποιώντας την βιβλιοθήκη `Glide`.

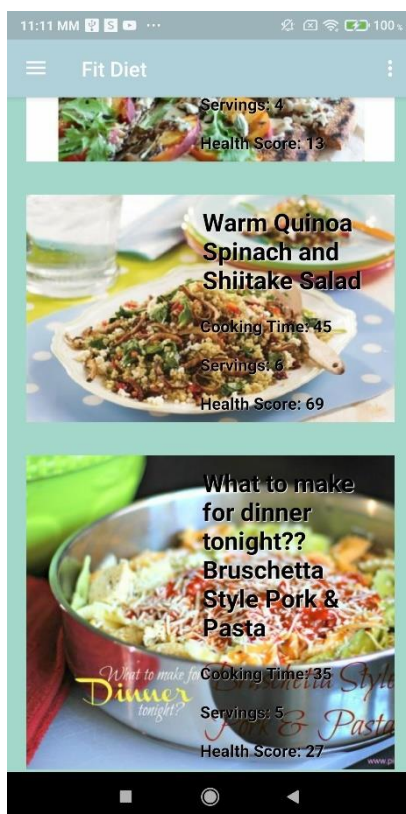
Επιπλέον, έχει χρησιμοποιηθεί η βοηθητική class `Recipe`, η οποία περιλαμβάνει διάφορες μεθόδους για την διαχείριση των δεδομένων της συνταγής. Ο `Constructor` `Recipe(String title, String healthScore, int servings, String image, String readyInMinutes, boolean veryHealthy)` αρχικοποιεί τα βασικά πεδία της συνταγής με τις τιμές που παρέχονται ως παράμετροι όπως ο τίτλος της συνταγής, το `healthScore`, η εικόνα καθώς και ο χρόνος μαγειρέματος και οι μερίδες της συνταγής. Επιπλέον, οι `Getters` και `Setters` που υπάρχουν για όλα τα πεδία, επιτρέπουν την πρόσβαση και την τροποποίηση των δεδομένων της συνταγής.

Για να μεταβεί ο χρήστης στην οθόνη που αφορά τις συνταγές, μπορεί να επιλέξει μια από τις 3 επιλογές (Πρωινό-Μεσημεριανό-Βραδινό) στην `Homepage-Mainactivity` ή από το `Navigation drawer` όπου υπάρχει `item` για τις συνταγές (`nav_Recipes`). Με την επιλογή ενός από τα `buttons` λαμβάνουμε συνταγές για το εκάστοτε γεύμα ενώ με το `item` «Συνταγές» ο χρήστης θα λάβει τυχαίες υγιεινές συνταγές.



Εικόνα 38. Homepage – «Υγιεινές συνταγές» - «Πρωινό Γεύμα-Μεσημεριανό-Βραδινό»

Όπως φαίνεται και στην παρακάτω εικόνα ο χρήστης επιλέγοντας κάτι από τα παραπάνω οδηγείται στην οθόνη των υγιεινών συνταγών, στις οποίες εμφανίζονται τα βασικά στοιχεία της συνταγής όπως ο τίτλος της συνταγής, το healthScore, η εικόνα καθώς και ο χρόνος μαγειρέματος και οι μερίδες της συνταγής. Εάν στη συνέχεια επιθυμεί, μπορεί να κάνει κλικ ώστε να δει και τα βήματα για την εκτέλεση της συνταγής.



Εικόνα 39. Οθόνη Συνταγές Recipes

5.2.13. Recipe

Η κλάση Recipe χρησιμοποιείται τόσο στην οθόνη για τις συνταγές όσο και στην οθόνη με την ανάλυση της συνταγής και περιλαμβάνει διάφορες μεθόδους για την διαχείριση των δεδομένων της συνταγής:

Constructor:

Ο κατασκευαστής Recipe(String title, String healthScore, int servings, String image, String readyInMinutes, 530olean veryHealthy) αρχικοποιεί τα βασικά πεδία τις συνταγής με τις τιμές που παρέχονται ως παράμετροι.

Getters και Setters:

Η κλάση περιλαμβάνει getters και setters για όλα τα πεδία, επιτρέποντας την πρόσβαση και την τροποποίηση των δεδομένων της συνταγής.

Για παράδειγμα:

- getTitle() και setTitle(String title) για τον τίτλο.

- `getCalories()` και `setCalories(double calories)` για τις θερμίδες.
- `getImageUrl()` και `setImageUrl(String image)` για την εικόνα.
- `getExtendedIngredients()` και `setExtendedIngredients(List<ExtendedIngredient> extendedIngredients)` για τα συστατικά.
- `getAnalyzedInstructions()` και `setAnalyzedInstructions(List<AnalyzedInstruction> analyzedInstructions)` για τις οδηγίες μαγειρέματος.

Εσωτερικές Κλάσεις:

Οι εσωτερικές κλάσεις `Nutrition`, `Nutrient`, `ExtendedIngredient`, και `AnalyzedInstruction` περιέχουν επιπλέον δεδομένα που σχετίζονται με τη συνταγή και περιλαμβάνουν επίσης `getters` και `setters` για την πρόσβαση και την τροποποίηση των δεδομένων τους.

- Η κλάση `Nutrition` περιέχει μια λίστα τύπου `Nutrient` που αναπαριστά τα θρεπτικά συστατικά.
- Η κλάση `Nutrient` περιέχει τον τίτλο, την ποσότητα και τη μονάδα μέτρησης κάθε θρεπτικού συστατικού.
- Η κλάση `ExtendedIngredient` περιέχει το όνομα, την ποσότητα και τη μονάδα μέτρησης κάθε συστατικού.
- Η κλάση `AnalyzedInstruction` περιέχει μια λίστα από `Step` που αναπαριστούν τα βήματα των οδηγιών μαγειρέματος.
- Η κλάση `Step` περιέχει τον αριθμό και την περιγραφή κάθε βήματος.

5.2.14. Οθόνη για Λεπτομέρειες συνταγής - `RecipeDetailsFragment`

Η κλάση `RecipeDetailsFragment` χρησιμοποιείται για την εμφάνιση των λεπτομερειών μιας συνταγής. Περιλαμβάνει διάφορες μεταβλητές για την ανάκτηση των στοιχείων της συνταγής και μεθόδους για την αρχικοποίηση αυτών των στοιχείων και την λήψη των απαραίτητων δεδομένων από ένα API.

Σταθερές και Μεταβλητές

Αρχικά, υπάρχουν διάφορες μεταβλητές που χρησιμοποιούνται για την ανάκτηση των στοιχείων της συνταγής. Η μεταβλητή `recipeImageView` ανακτά από το `api` την εικόνα της συνταγής, η `titleTextView` τον τίτλο της συνταγής, `getReadyInMinutes` το χρόνο μαγειρέματος, η `servingsTextView` τον αριθμό των μερίδων, και τέλος η `healthScoreTextView` τον δείκτη υγείας της συνταγής. Επιπλέον, η `ingredientsTextView` ανακτά τη λίστα με τα συστατικά και η `instructionsTextView` ανακτά από το `api` τις οδηγίες μαγειρέματος.

Η class `Recipe` μας δίνει τη μεταβλητή `recipe` για όλα τα δεδομένα της συνταγής. Η μεταβλητή `api` είναι ένα αντικείμενο της `RecipeApi` που χρησιμοποιείται για την επικοινωνία με το API.

Μέθοδοι

Η κλάση `RecipeDetailsFragment` περιλαμβάνει διάφορες μεθόδους για την αρχικοποίηση και διαχείριση των δεδομένων της συνταγής. Η μέθοδος `newInstance(Recipe recipe)` δημιουργεί ένα νέο instance του `RecipeDetailsFragment` και περνά τη συνταγή μέσω ενός `Bundle`. Αυτή η μέθοδος χρησιμοποιείται για τη δημιουργία νέων instances του fragment με τα δεδομένα της συνταγής.

Η μέθοδος `onCreate(Bundle savedInstanceState)` καλείται κατά τη δημιουργία του fragment και χρησιμοποιείται για την αρχικοποίηση των βασικών στοιχείων του fragment. Ελέγχει αν υπάρχουν arguments που περιέχουν τα δεδομένα της συνταγής και δημιουργεί ένα instance του Retrofit για την επικοινωνία με το API. Η μέθοδος `onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)` καλείται για να δημιουργήσει την εμφάνιση του fragment. Αρχικοποιεί τα views και τις μεταβλητές της συνταγής και καλεί τις μεθόδους `fetchInstructions` και `fetchIngredients` για να λάβει τα δεδομένα της συνταγής από το API και να τα εμφανίσει στα αντίστοιχα views.

Η μέθοδος `fetchInstructions(int recipeId, TextView instructionsTextView)` καλείται για να λάβει τις οδηγίες μαγειρέματος της συνταγής από το API. Χρησιμοποιεί το Retrofit για να κάνει μια αίτηση στο API και να λάβει τις οδηγίες, τις οποίες στη συνέχεια εμφανίζει στο `instructionsTextView`. Η μέθοδος `fetchIngredients(int recipeId, TextView ingredientsTextView)` καλείται για να λάβει τα συστατικά της συνταγής από το API. Χρησιμοποιεί το Retrofit για να κάνει μια αίτηση στο API και να λάβει τα συστατικά, τα οποία στη συνέχεια εμφανίζει στο `ingredientsTextView`.

Η κλάση `RecipeDetailsResponse` χρησιμοποιείται για την αποθήκευση των δεδομένων που λαμβάνονται από το API. Περιέχει τη λίστα των βημάτων (`steps`) και τη λίστα των συστατικών (`ingredients`) της συνταγής. Κάθε βήμα περιλαμβάνει τον αριθμό του βήματος (`number`) και την περιγραφή του (`step`). Η κλάση `Ingredient` χρησιμοποιείται για την αποθήκευση των στοιχείων κάθε συστατικού, όπως το όνομα (`name`), την ποσότητα (`amount`) και τη μονάδα μέτρησης (`unit`).

Με αυτές τις μεταβλητές και μεθόδους, η κλάση `RecipeDetailsFragment` μπορεί να διαχειριστεί πλήρως τις λεπτομέρειες μιας συνταγής, επιτρέποντας την εύκολη ανάκτηση, αποθήκευση και προβολή των δεδομένων της συνταγής στον χρήστη.



Εικόνα 40. Οθόνη Λεπτομέρειες Συνταγής RecipeDetailsFragment

5.2.15. Οθόνη για Πλάνο διατροφής - MealPlanFragment

Η κλάση MealPlanFragment εμφανίζει ένα ημερήσιο πλάνο γευμάτων και τα διατροφικά στοιχεία που περιέχει. Αυτή η κλάση δημιουργεί και διαχειρίζεται την εμφάνιση των γευμάτων με βάση τις θερμίδες στόχου που ορίζει ο χρήστης. Σε περίπτωση που μεταβεί στη συγκεκριμένη οθόνη, από τη MainActivity, πριν καταχωρήσει τα στοιχεία του όπως ηλικία, ύψος, βάρος κτλ. για να μπορέσουν να υπολογιστούν οι απαραίτητες θερμίδες που χρειάζεται ημερησίως για να επιτύχει το στόχο του, καθορίζεται ένας default στόχος θερμίδων ορισμένος στις 2000 kcal.

Μεταβλητές και Σταθερές

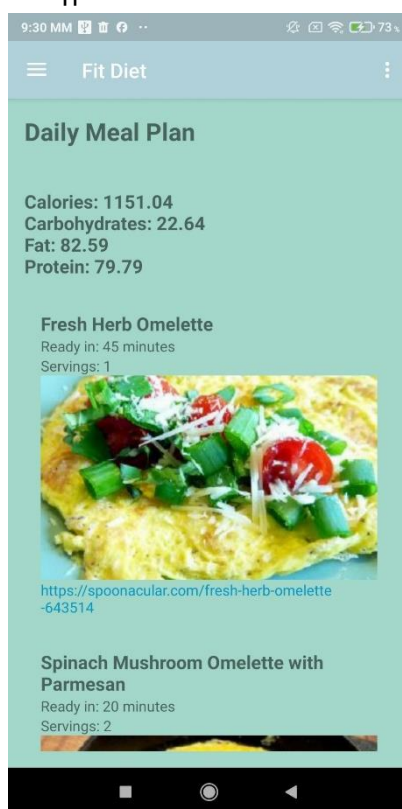
Στην `MealPlanFragment`, ορίζονται μεταβλητές για την προβολή του πλάνου γευμάτων (`mealPlanTextView`), των διατροφικών στοιχείων (`nutrientsTextView`), καθώς και η μεταβλητή που ορίζονται οι θερμίδες στόχου (`targetCaloriesString`). Επιπρόσθετα, υπάρχει μια σταθερά (`ARG_TARGET_CALORIES`) που χρησιμοποιείται για τη μεταφορά των θερμίδων στόχου ανάμεσα στα `Fragments`.

Μέθοδοι

Η μέθοδος `onCreateView` είναι υπεύθυνη για τη δημιουργία του `layout` του `Fragment` και την αρχικοποίηση των `UI` στοιχείων, καθώς και για την κλήση της `getDailyMealPlan` για την ανάκτηση του πλάνου γευμάτων αν οι θερμίδες στόχου είναι διαθέσιμες.

Η `getDailyMealPlan` χρησιμοποιεί το `Retrofit` για να κάνει ένα `API call` προς το `Spoonacular API` για να ανακτήσει το ημερήσιο πλάνο γευμάτων. Αν η ανάκτηση είναι επιτυχής, καλεί τη μέθοδο `displayMealPlan` για να εμφανίσει τα δεδομένα.

Η `displayMealPlan` αναλαμβάνει την προβολή των γευμάτων και των θρεπτικών συστατικών. Για κάθε γεύμα, δημιουργούνται δυναμικά `UI` στοιχεία και προστίθενται στον κατάλληλο `container`. Επιπλέον, χρησιμοποιείται η βιβλιοθήκη `Glide` για τη φόρτωση και εμφάνιση των εικόνων των γευμάτων. Εάν ο χρήστης κάνει κλικ στο `url` (`sourceUrlTextView`), τότε ανοίγει το `URL` της συνταγής στον προεπιλεγμένο `browser`.



Εικόνα 41. Οθόνη Ημερήσιου πλάνου διατροφής – `MealPlanFragment`



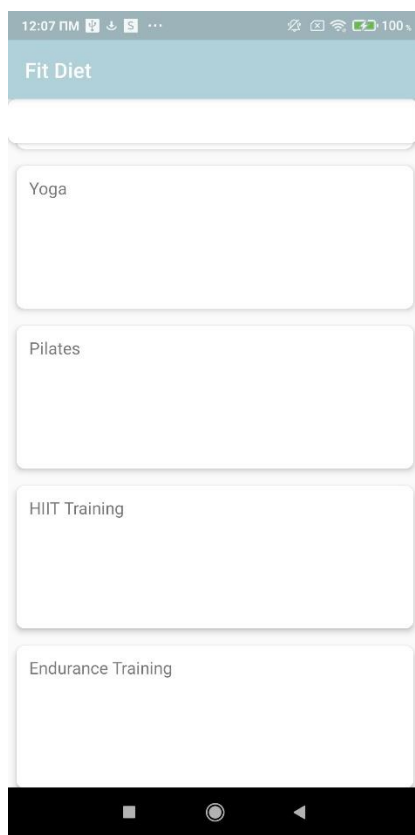
Εικόνα 42. Σελίδα Spoonacular με επιλεγμένη συνταγή.

5.2.16. Οθόνη για Λίστα Κατηγοριών Ασκήσεων - LearnActivity

Η κλάση LearnActivity είναι μια Activity που χρησιμοποιείται για την εμφάνιση μιας λίστας με κατηγορίες ασκήσεων. Αυτή η δραστηριότητα περιλαμβάνει ένα RecyclerView για την εμφάνιση της λίστας, και μια κάρτα (CardView) για κάθε στοιχείο της λίστας. Ο κώδικας περιλαμβάνει επίσης μια στατική εσωτερική κλάση YoutubeAPI για την αλληλεπίδραση με το YouTube API.

Κατά την αρχικοποίηση της δραστηριότητας στη μέθοδο onCreate, ορίζεται το layout και αρχικοποιούνται οι μεταβλητές. Μια λίστα από κατηγορίες ασκήσεων δημιουργείται και προστίθεται στη λίστα κατηγοριών. Το RecyclerView και η κάρτα αρχικοποιούνται με τη μέθοδο findViewById. Το RecyclerView ρυθμίζεται να χρησιμοποιεί έναν LinearLayoutManager, και ο προσαρμογέας LearnListAdapter χρησιμοποιείται για να προσαρμόσει τα δεδομένα στο RecyclerView.

Όπως φαίνεται στην παρακάτω εικόνα, ο χρήστης μπορεί να επιλέξει μεταξύ διαφόρων κατηγοριών ασκήσεων όπως Before Workout Stretches, Yoga, Pilates, HIIT Training κτλ.. Με βάση αυτήν την επιλογή, οδηγείται στην επόμενη Activity (Exercise) στην οποία βλέπει μια λίστα μόνο με τις ασκήσεις της συγκεκριμένης κατηγορίας.



Εικόνα 43. Οθόνη Λίστα Κατηγοριών Ασκήσεων - LearnActivity

5.2.17. YoutubeAPI

Η βοηθητική κλάση YoutubeAPI περιλαμβάνει σταθερές που σχετίζονται με τα αιτήματα προς το YouTube API, όπως η βάση URL, τα μέρη του URL για την αναζήτηση, τα μέγιστα αποτελέσματα, η σειρά των αποτελεσμάτων και ο τύπος των δεδομένων. Επίσης, περιέχει σταθερές που καθορίζουν τους όρους αναζήτησης για κάθε κατηγορία ασκήσεων, όπως "Stretching After Workout", "Yoga" κτλ.

Η διεπαφή VideoAPI περιλαμβάνει μια μέθοδο για να κάνει αιτήσεις GET στο YouTube API. Η μέθοδος `getVideoAPI` δημιουργεί και επιστρέφει ένα αντικείμενο VideoAPI χρησιμοποιώντας το Retrofit, εξασφαλίζοντας ότι το αντικείμενο δημιουργείται μόνο μία φορά (singleton pattern).

Αυτή η δομή επιτρέπει στη δραστηριότητα LearnActivity να ανακτά δεδομένα από το YouTube API και να τα εμφανίζει σε έναν RecyclerView μέσω ενός Adapter (LearnListAdapter). Μέσω αυτής της διαδικασίας, οι χρήστες μπορούν να βλέπουν και να επιλέγουν από μια λίστα με διάφορες κατηγορίες ασκήσεων, κάνοντας την εφαρμογή πιο διαδραστική και χρήσιμη για την εκμάθηση και εκτέλεση ασκήσεων.

5.2.18. Οθόνη με Λίστα για συγκεκριμένη κατηγορία ασκήσεων - Exercise

Η κλάση `Exercise` είναι μια Activity που χρησιμοποιείται για την εμφάνιση μιας λίστας με βίντεο ασκήσεων από το YouTube. Η κλάση περιλαμβάνει διάφορες μεταβλητές και μεθόδους για την αρχικοποίηση αυτών των στοιχείων, τη λήψη δεδομένων από το YoutubeAPI και την εμφάνισή τους στον χρήστη.

Σταθερές και Μεταβλητές

Αρχικά, η κλάση περιλαμβάνει μεταβλητές όπως RecyclerView που χρησιμοποιείται για την εμφάνιση της λίστας των βίντεο, τις λίστες Workout allWorkouts και filteredWorkouts, για όλες τις ασκήσεις και φιλτραρισμένες ασκήσεις και τη λίστα VideoYT tutorials` με τα αντίστοιχα βίντεο ασκήσεων.

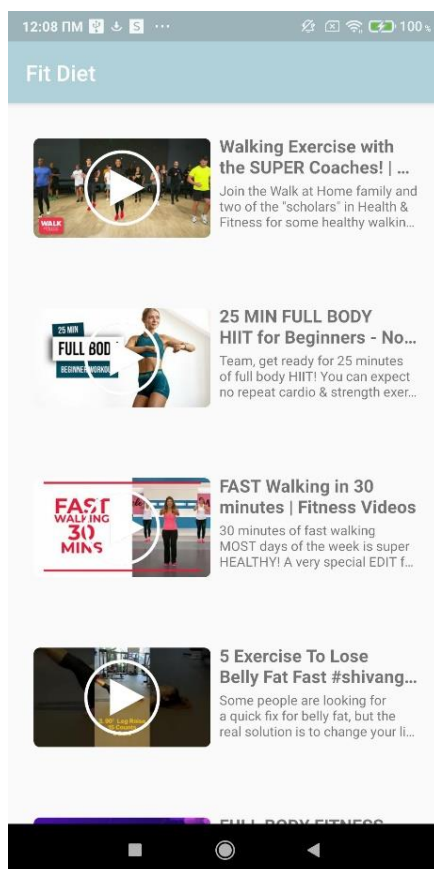
Επιπλέον, υπάρχει ένα σταθερό κλειδί API (`API_KEY`) που χρησιμοποιείται για την πρόσβαση στο YouTube API.

Μέθοδοι

Η κλάση περιλαμβάνει τη μέθοδο onCreate, που αρχικοποιεί το layout της δραστηριότητας και τις μεταβλητές της κλάσης, δημιουργεί το RecyclerView και τον προσαρμογέα και καλεί τη μέθοδο `getJSON` αν η λίστα `tutorials` είναι κενή.

Η μέθοδος `getJSON()` χρησιμοποιείται για τη λήψη δεδομένων από το YouTubeAPI. Αρχικά, λαμβάνει την κατηγορία που έχει περαστεί ως extra από το Intent. Στη συνέχεια, κατασκευάζει το URL για την αίτηση στο API με βάση την κατηγορία αυτή. Ανάλογα με την κατηγορία, κατασκευάζει το αντίστοιχο URL για την αίτηση και στη συνέχεια εκτελεί την αίτηση χρησιμοποιώντας το Retrofit.

Η βοηθητική class «JSON_TutorialResponseModel» χρησιμοποιείται για την αποθήκευση της απάντησης από το YouTubeAPI. Περιλαμβάνει τις μεταβλητές nextPageToken (token για την επόμενη σελίδα των αποτελεσμάτων) & items (η λίστα με αντικείμενα `VideoYT` που αντιπροσωπεύουν τα βίντεο), και περιλαμβάνει τους αντίστοιχους getters και setters



Εικόνα 44. Οθόνη Λίστα Ασκήσεων συγκεκριμένης κατηγορίας- Exercise

5.2.19. Οθόνη -Tutorial άσκησης από το YouTube - TutorialPlayback

Η κλάση TutorialPlayback είναι υπεύθυνη για την αναπαραγωγή βίντεο από το YouTube και την εμφάνιση των πληροφοριών του βίντεο, όπως ο τίτλος, η περιγραφή και η ημερομηνία δημοσίευσης. Αυτή η δραστηριότητα επεκτείνει την YouTubeBaseActivity και χρησιμοποιεί την YouTubePlayerView για την αναπαραγωγή του βίντεο.

Σταθερές και Μεταβλητές

Οι μεταβλητές youtubePlayerView, tvTitle, tvDescription και tvPublishedAt αρχικοποιούνται για να αναπαραστήσουν την προβολή του YouTube player και τα στοιχεία κειμένου για τον τίτλο, την περιγραφή και την ημερομηνία δημοσίευσης αντίστοιχα.

Μέθοδοι

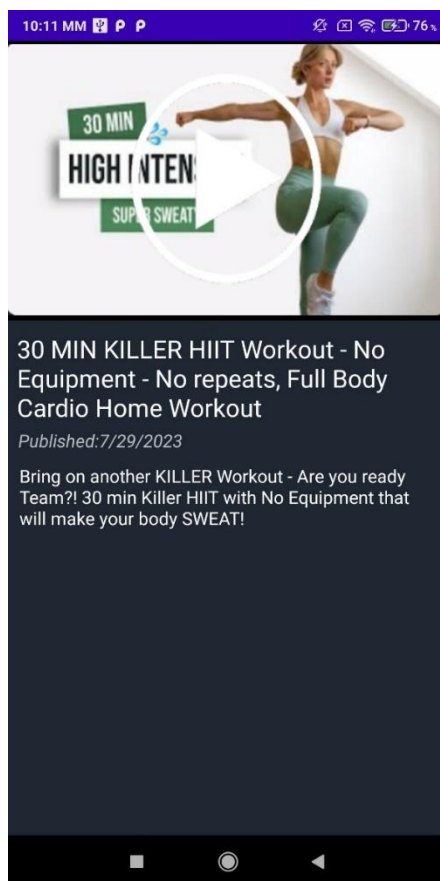
Αρχικά, μέσα στη μέθοδο onCreate, γίνεται η αρχικοποίηση της διάταξης (layout) καθώς επίσης και των αντίστοιχων στοιχείων UI.

Οι πληροφορίες του βίντεο ανακτώνται από ένα αντικείμενο VideoYT, το οποίο περνά ως παράμετρος μέσω του Intent. Αυτές οι πληροφορίες περιλαμβάνουν τον τίτλο, την περιγραφή, το URL του βίντεο και την ημερομηνία δημοσίευσης. Η ημερομηνία δημοσίευσης μετατρέπεται σε αναγνώσιμη μορφή (MM/dd/yyyy) χρησιμοποιώντας την κλάση SimpleDateFormat.

Η αρχικοποίηση της YouTubePlayerView πραγματοποιείται καλώντας τη μέθοδο initialize με το API key για το YouTube. Εάν η αρχικοποίηση είναι επιτυχής, το βίντεο φορτώνεται και αναπαράγεται μέσω του youtubePlayer. Εάν η αρχικοποίηση αποτύχει, εμφανίζεται ένα μήνυμα σφάλματος.

Τέλος, οι πληροφορίες του βίντεο εμφανίζονται στα αντίστοιχα στοιχεία UI (TextView). Οι τίτλοι, οι περιγραφές και η μορφοποιημένη ημερομηνία δημοσίευσης ανατίθενται στα tvTitle, tvDescription και tvPublishedAt, αντίστοιχα.

Αυτή η προσέγγιση διασφαλίζει ότι η δραστηριότητα TutorialPlayback μπορεί να αναπαράγει βίντεο από το YouTube και να εμφανίζει τις σχετικές πληροφορίες με έναν εύχρηστο και οπτικά ευχάριστο τρόπο, χρησιμοποιώντας το API του YouTube και τα διαθέσιμα εργαλεία της Android πλατφόρμας. Παρακάτω υπάρχει η σχετική εικόνα που βλέπει ο χρήστης όταν μπαίνει στην οθόνη.



Εικόνα 45. Οθόνη Tutorial άσκησης από το YouTube – TutorialPlayback

Σε περίπτωση που πατήσει το play μπορεί να δει το επιθυμητό Tutorial.

5.2.20. Οθόνη Σχετικά με εμάς - AboutFragment

Η class AboutFragment είναι υπεύθυνη για την προβολή πληροφοριών σχετικά με την εφαρμογή και παρέχει συνδέσμους προς εξωτερικές ιστοσελίδες μέσω εικόνων. Ακολουθεί μια ανάλυση της λειτουργικότητας της κλάσης:

Μέθοδοι

Η μέθοδος onCreateView καλείται όταν το fragment δημιουργείται και είναι υπεύθυνη για την αρχικοποίηση του UI και την ορισμό των λειτουργιών των κουμπιών. Κατά τη δημιουργία της όψης (view) του fragment, η μέθοδος onCreateView φουσκώνει (inflate) το layout fragment_about και επιστρέφει την προβολή (view). Μέσα σε αυτή τη μέθοδο, βρίσκονται οι αναφορές σε συγκεκριμένα στοιχεία UI, όπως τα ImageView για το Facebook και της σελίδας του Πανεπιστημίου Πειραιά.

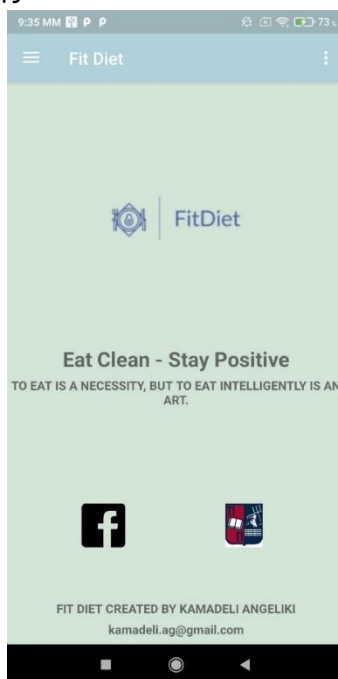
Java

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_about,
container, false);

    ImageView facebook = view.findViewById(R.id.imageView7);
    facebook.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                Intent fbIntent = new Intent(Intent.ACTION_VIEW);
                fbIntent.setData(Uri.parse("https://el-
gr.facebook.com/PMSPliroforiki/"));
                startActivity(fbIntent);
            } catch (Exception e) {}
        }
    });

    ImageView twitter = view.findViewById(R.id.imageView8);
    twitter.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                Intent tIntent = new Intent(Intent.ACTION_VIEW);
                tIntent.setData(Uri.parse("https://www.cs.unipi.gr/index.php?option
=com_k2&view=item&layout=item&id=61&Itemid=627&lang=el"));
                startActivity(tIntent);
            } catch (Exception e) {}
        }
    });
    return view;
}
```

Στη συνέχεια, οι δύο `ImageView` αρχικοποιούνται και προστίθενται `OnClickListeners` για κάθε μία. Όταν ο χρήστης κάνει κλικ στις εικόνες, ξεκινά μια πρόθεση (`intent`) για να ανοίξει την αντίστοιχη ιστοσελίδα στον προεπιλεγμένο περιηγητή (`browser`) του χρήστη. Αν προκύψει κάποιο σφάλμα κατά την προσπάθεια ανοίγματος της ιστοσελίδας, το σφάλμα αγνοείται με το `catch` (`Exception e`). Αυτή η μέθοδος διασφαλίζει ότι οι χρήστες μπορούν εύκολα να έχουν πρόσβαση σε εξωτερικές πληροφορίες σχετικά με την εφαρμογή ή τον οργανισμό, ενισχύοντας την εμπειρία χρήσης.



Εικόνα 46. AboutFragment.

Όταν επιλεγθεί το `button` του Facebook, ο χρήστης ανοίγει στη σελίδα του Πανεπιστημίου Πειραιά στο Facebook, ενώ με το `button` που έχει το λογότυπο του Πανεπιστημίου Πειραιά στην αντίστοιχη ιστοσελίδα στον προεπιλεγμένο περιηγητή (`browser`) του χρήστη.



Εικόνα 47. Site Πανεπιστημίου Πειραιά.



Εικόνα 48. Σελίδα Facebook Πανεπιστημίου Πειραιά.

6. Συμπεράσματα και Μελλοντική Εργασία

6.1. Συμπεράσματα:

Η εφαρμογή FitDiet συνιστά ένα αξιόπιστο και καινοτόμο εργαλείο για τη βελτίωση της υγείας και της ευεξίας των χρηστών της. Μέσω της ενσωμάτωσης επιστημονικά τεκμηριωμένων δεδομένων και πρακτικών, η FitDiet παρέχει στους χρήστες εξατομικευμένες συμβουλές διατροφής και γυμναστικής, οι οποίες ανταποκρίνονται στις ατομικές τους ανάγκες και προτιμήσεις. Η εφαρμογή διευκολύνει την παρακολούθηση της προόδου των χρηστών μέσω μιας διαδραστικής διεπαφής που ενθαρρύνει την ενεργή συμμετοχή και την ενσωμάτωση υγιεινών συνθηκών στην καθημερινή ζωή.

Επιπλέον, η ευχρηστία και η ευελιξία της εφαρμογής την καθιστούν προσιτή σε ένα ευρύ φάσμα χρηστών, ανεξαρτήτως ηλικίας, φυσικής κατάστασης ή επιπέδου γνώσης στον τομέα της διατροφής και της γυμναστικής. Η δυνατότητα παρακολούθησης της ημερήσιας πρόσληψης θερμίδων και νερού, καθώς και η υποστήριξη με λεπτομερείς οδηγίες και βίντεο προπονήσεων, προσδίδουν μια ολοκληρωμένη εμπειρία στους χρήστες, ενισχύοντας την αφοσίωσή τους στην επίτευξη των προσωπικών τους στόχων.

Συνοψίζοντας, η εφαρμογή FitDiet προσφέρει μια ολοκληρωμένη λύση για τη βελτίωση της φυσικής κατάστασης και της διατροφής, βασισμένη σε επιστημονικά δεδομένα και πρακτικές. Η επιτυχία της έγκειται στη δυνατότητά της να προσφέρει εξατομικευμένες συμβουλές και να προσαρμόζεται στις ανάγκες των χρηστών, προάγοντας έναν υγιεινό και ισορροπημένο τρόπο ζωής.

6.2. Περιορισμοί και Προτάσεις Βελτίωσης

Παρά την αξιόλογη προσφορά της FitDiet, υπάρχουν ορισμένοι περιορισμοί και προκλήσεις που θα πρέπει να αντιμετωπιστούν για την περαιτέρω βελτίωση της εφαρμογής. Πρώτον, ο αλγόριθμος σύστασης της εφαρμογής, αν και βασισμένος σε επιστημονικά δεδομένα, μπορεί να επωφεληθεί από περαιτέρω ανάπτυξη και βελτιώσεις. Η αυξημένη εξατομίκευση των συμβουλών, με την ενσωμάτωση περισσότερων παραμέτρων όπως το επίπεδο φυσικής δραστηριότητας, οι διατροφικές προτιμήσεις και οι αλλεργίες, θα μπορούσε να προσδώσει μεγαλύτερη αξία στους χρήστες.

Επιπλέον, η αύξηση της ποικιλίας των διατροφικών επιλογών και των προπονήσεων αποτελεί σημαντικό πεδίο βελτίωσης. Η προσθήκη νέων και διαφοροποιημένων ασκήσεων, καθώς και η ενσωμάτωση εξειδικευμένων προγραμμάτων διατροφής, θα ενίσχυαν την ελκυστικότητα και την αποτελεσματικότητα της εφαρμογής. Η δημιουργία μιας κοινότητας χρηστών, μέσω της οποίας θα μπορούσαν να ανταλλάσσονται εμπειρίες, συμβουλές και κίνητρα, θα μπορούσε να προσδώσει μια διαδραστική διάσταση στην εμπειρία χρήσης της εφαρμογής.

Τέλος, η διασύνδεση της εφαρμογής με άλλες πλατφόρμες και συσκευές παρακολούθησης φυσικής δραστηριότητας θα μπορούσε να ενισχύσει την ακρίβεια των δεδομένων και τη συνολική εμπειρία του χρήστη. Η FitDiet θα μπορούσε να επωφεληθεί από την ενσωμάτωση δεδομένων από συσκευές όπως τα smartwatches και οι αισθητήρες φυσικής

δραστηριότητας, παρέχοντας μια πιο ολοκληρωμένη εικόνα της υγείας και της ευεξίας του χρήστη.

Σε γενικές γραμμές, οι προτάσεις αυτές αποσκοπούν στην περαιτέρω βελτίωση και επέκταση της εφαρμογής FitDiet, ώστε να συνεχίσει να προσφέρει αξιόπιστες και αποτελεσματικές λύσεις στους χρήστες της, προωθώντας τη βελτίωση της υγείας και της ευεξίας τους.

Βιβλιογραφία

- Παγκόσμιος Οργανισμός Υγείας, 2015. Κατευθυντήριες Γραμμές για τη Σωματική Δραστηριότητα, s.l.: Παγκόσμιος Οργανισμός Υγείας.
- Committee, D. G. A., 2015. Scientific Report of the 2015 Dietary Guidelines Advisory Committee, s.l.: Office of Disease Prevention and Health Promotion.
- Medicine, A. C. o. S., 2014. ACSM's Guidelines for Exercise Testing and Prescription. s.l.:Lippincott Williams & Wilkins.
- Mifflin, M. D. e. a., 1990. A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition*, 51(2), pp. 241-247.
- Mifflin, M. D. S. J. S. T. H. L. A. S. B. J. D. S. A. & K. Y. O., 1990. A new predictive equation for resting energy expenditure in healthy individuals. *The American Journal of Clinical Nutrition*, 51(2), pp. 241-247.
- Nestle, M., 2013. Food Politics: How the Food Industry Influences Nutrition and Health. s.l.:University of California Press.
- Roza, A. M. & S. H. M., 1984. The Harris Benedict equation reevaluated: resting energy requirements and the body cell mass. *The American Journal of Clinical Nutrition*, 40(1), pp. 168-182.
- Swinburn, B. e. a., 2011. The global obesity pandemic: shaped by global drivers and local environments. *The Lancet*, 27 08, 378(9793), pp. 804-814.
- Tessier, S. e. a., 2008. Analysis of dietary data from food diaries. *Epidemiologic Methods*, 1(1), pp. 107-113.
- Thompson, F. E. & S. A. F., 2013. Dietary assessment methodology. Στο: 1 επιμ. s.l.:s.n., pp. 3-39.
- U.S. Department of Health and Human Services, 2020. [Ηλεκτρονικό] Available at: <https://www.dietaryguidelines.gov>
- World Health Organization, 2020. Healthy diet. [Ηλεκτρονικό] Available at: <https://www.who.int/news-room/fact-sheets/detail/healthy-diet> [Πρόσβαση 24 April 2020].
- Ακαδημία Διατροφής και Διαιτολογίας, 2017. Η σημασία της συνεργασίας διαιτολόγων και τεχνολογίας. *Περιοδικό Διατροφικής Θεραπείας*, 15(2), pp. 56-67.
- Ινστιτούτο Τεχνολογίας Υγείας, 2021. Η χρήση φορητών συσκευών στην παρακολούθηση της υγείας. *Τεχνολογία Υγείας*, 1(3), pp. 78-89.
- Μπεχράκης, Π., 2002. Διατροφή και Υγεία. s.l.:Εκδόσεις Παπαζήση.
- Πάππας, Γ. & Κ. Β., 2016. Βασικός Μεταβολικός Ρυθμός και Θερμίδες. *Περιοδικό Κλινικής Διατροφής*, 12(1), pp. 34-45.
- Τμήμα Διαιτολογίας και Διατροφής, Χ. Π., 2020. Εξατομικευμένες Διατροφικές Προτάσεις και Συμμόρφωση, s.l.: Χαροκόπειο Πανεπιστήμιο.
- Υπουργείο Υγείας και Κοινωνικής Αλληλεγγύης, 2012. Διατροφικές Οδηγίες για Ενήλικες, s.l.: Υπουργείο Υγείας και Κοινωνικής Αλληλεγγύης.
- Alepis, E., & Patsakis, C. (2019). Unravelling Security Issues of Runtime Permissions in Android. *Journal of Hardware and Systems Security*, 3(1), 45-63.
- Torres, J., de los Santos, S., Alepis, E., & Patsakis, C. (2019). Behavioral Biometric Authentication in Android Unlock Patterns through Machine Learning. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 146-154).

Kontogianni, A., Alepis, E., Virvou, M., & Patsakis, C. (2024). Smart Tourism-The Impact of Artificial Intelligence and Blockchain. In Intelligent Systems Reference Library (Vol. 249, pp. 1-178). Springer. ISBN 978-3-031-50882-0.

Politou, E. A., Alepis, E., Virvou, M., & Patsakis, C. (2022). Privacy and Data Protection Challenges in the Distributed Era. Springer. ISBN 978-3-030-85442-3.

Virvou, M., Alepis, E., Tsihrintzis, G. A., & Jain, L. C. (2020). Machine Learning Paradigms - Advances in Learning Analytics. In Machine Learning Paradigms (pp. 1-5).

Houlis, C., Patsakis, C., & Alepis, E. (2019). Smart Android Application using Self-Destructive Identities against Cyber Harassment. In Proceedings of the International Conference on Information, Intelligence, Systems and Applications (IISA) (pp. 1-7).