

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ - ΛΟΓΙΣΜΙΚΟ**  
**ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ**

**Παραδοτέο από:**

ΥΦΑΝΤΗΣ ΓΙΩΡΓΟΣ,  
ΤΟΥΛΟΥΠΗΣ ΓΙΑΝΝΗΣ,  
ΦΩΤΟΠΟΥΛΟΣ ΓΙΑΝΝΗΣ,

**A.M.:**

Π18163,  
Π18217,  
Π18238

**Επιβλέπον Καθηγητής:** Ευθύμιος Αλέπης

- **Εισαγωγή**
  - Εκφώνηση
  - Ανάλυση Απαιτήσεων
  - Βιβλιοθήκες
- **Διαγράμματα**
  - Διάγραμμα Τάξεων
  - Διάγραμμα Περιπτώσεων Χρηστών
- **Ανάλυση UI**
  - Launch Activity
  - Login Activity
  - Register Activity
  - Quiz Activity
  - Diary Fragment
    - Add Meal Activity
    - Add Exercise Activity
  - Map Fragment
  - Stats Fragment
  - Goals Fragment
  - Account Fragment
  - Settings Activity
  - App Lock
    - App Lock Register Activity
    - App Lock Activity
- **Ανάλυση Κώδικα**

- Launch Activity
- Login Activity
- Register Activity
- Quiz Activity
- Diary Fragment
  - Add Meal Activity
  - Add Exercise Activity
- Map Fragment
- Stats Fragment
- Goals Fragment
- Account Fragment
- Settings Activity
- App Lock
  - App Lock Register Activity
  - App Lock Activity
- Helper Classes
  - Notification Receiver
  - Recommended Values Helper
- Application

# 1. Εισαγωγή

## Εκφώνηση

Ανάπτυξη Android εφαρμογής καταγραφής και ειδοποίησης σε σχέση με τις ημερήσιες θερμίδες που καταναλώνει ένας χρήστης. Χρειαζόμαστε καταγραφή γευμάτων, στατιστικά, στόχους και αντίστοιχες ειδοποιήσεις.

# Ανάλυση Απαιτήσεων

Ο χρήστης θα πρέπει να μπορεί:

- Να δημιουργήσει λογαριασμό με το email του και ένα κωδικό ασφαλείας.
- **Έξτρα: Θα μπορεί να δημιουργήσει λογαριασμό και με Google και Twitter**
- Να μπορεί να συνδεθεί σε υπάρχον λογαριασμό
- **Έξτρα: Θα μπορεί να συνδεθεί και με Google και Twitter**
- Να εισάγει δεδομένα όπως το βάρος, ύψος και την ηλικία του.
- Για κάθε μέρα να μπορεί να δει τις θερμίδες που έχει καταναλώσει και κάψει, καθώς και πόσα μακροθρεπτικά συστατικά έλαβε εκείνη την μέρα.
- Για κάθε μέρα να μπορεί να προσθέσει τα γεύματα και την γυμναστική που έχει κάνει μέσα στην ημέρα.
- **Εξτρά: Ο χρήστης θα μπορεί να προσθέσει και τα δικά του φαγητά και ασκήσεις, τα οποία θα αποθηκεύονται τοπικά στην συσκευή.**
- Να μπορεί να δει στατιστικά όπως την αναλογία των μακροθρεπτικών συστατικών στην διατροφή του, η εξέλιξη του βάρους του κ.α.
- **Εξτρά: Τα στατιστικά θα τα βλέπει σε μορφή διαγραμμάτων, και θα μπορεί να επιλέξει χρονική περίοδο (ανά εβδομάδα, μήνα, τρίμηνο)**
- Να μπορεί να προσθέσει στόχους τους οποίους θα μπορεί να τους σημειώσει ως ολοκληρωμένους όταν τους καταφέρει.
- Να λαμβάνει ειδοποιήσεις για να του υπενθυμίζει να χρησιμοποιήσει την εφαρμογή.

Επιπλέον η εφαρμογή θα πρέπει να περιέχει χρήση τουλάχιστον 2 αισθητήρων και του αισθητήρα GPS. Πιο συγκεκριμένα η εφαρμογή κάνει την εξής χρήση των παρακάτω αισθητήρων:

- **Αισθητήρας GPS και Type\_Step\_Counter.**

- Η αξιοποίηση αυτών των αισθητήρων γίνεται στο Map Fragment στο οποίο ο χρήστης θα πρέπει να χορηγήσει απαραίτητα άδεια χρήσης του αισθητήρα GPS και του Activity Recognition. Αφού χορηγήσει άδεια χρήσης το Fragment αυτό περιέχει έναν χάρτη ο οποίος θα ακολουθεί την τοποθεσία του χρήστη.
- Επιπλέον σε αυτό το Fragment θα υπάρχει ένα κουμπί “Εκκίνηση”. Στο πάτημα αυτού του κουμπιού θα ξεκινάει ένα χρονόμετρο, ο βηματομετρητής καθώς θα αρχίσουμε να παίρνουμε δεδομένα όπως η ταχύτητα του χρήστη, τις θερμίδες που καίει καθώς και στον χάρτη θα αρχίζει να σημειώνει την διαδρομή που έχει κάνει.
- Η διαδικασία αυτή θα σταματάει είτε όταν πατήσουμε το κουμπί “Παύση”, σε αυτή την περίπτωση ο χρήστης θα μπορεί να συνεχίσει από εκεί που σταμάτησε αν ξαναπατήσει το κουμπί “Εκκίνηση”, ή αν κλείσει το Fragment, σε αυτή την περίπτωση η διαδικασία θα ξαναρχίσει από την αρχή όταν το θελήσει ο χρήστης.

- **Αισθητήρα Bio-metrics – Fingerprint Scanner**

- Η αξιοποίηση αυτού του αισθητήρα μπορεί να γίνει μέσω του Activity – Λειτουργία “App Lock” το οποίο ενεργοποιείτε όταν η εφαρμογή μπει στο παρασκήνιο για τουλάχιστον 10 δευτερόλεπτα. Αυτή η λειτουργία εμποδίζει τον χρήστη να χρησιμοποιήσει την εφαρμογή εκτός και αν ξεκλειδώσει την εφαρμογή είτε με έναν κωδικό που έχει θέσει αυτός είτε με την χρήση του δαχτυλικού αποτυπώματος.

Επιπλέον σε πολλά Activities και Fragments της εφαρμογής

μπορεί να γίνει χρήση του μικροφώνου. Πιο συγκεκριμένα χρήση του μικροφώνου μπορεί να γίνει στα εξής:

- Add Food Activity
- Add Exercise Activity
- Goals Fragment

Σε όλα απο τα παραπάνω η χρήση του μικροφώνου μπορεί να γίνει για την αναζήτηση αντικειμένων στην αντίστοιχη σελίδα (φαγητό , άσκηση, στόχος και κατορθώματα).

## Βιβλιοθήκες

- **'androidx.biometric:biometric:1.1.0'**
- Για την χρήση του δαχτυλικού αποτυπώματος.
- **'androidx.lifecycle:lifecycle-process:2.4.1'**
- Για μπορούμε να γνωρίζουμε πότε η εφαρμογή πήγε στο παρασκήνιο.
- **'com.github.AnyChart:AnyChart-Android:1.1.2'**
- Για την παρουσίαση των στατιστικών σε μορφή διαγραμμάτων.
- **'com.google.android.gms:play-services-maps:18.0.2'**
- Για την χρήση Google Maps
- **'com.google.code.gson:gson:2.9.0'**
- Για την προσθήκη αντικειμένων στα Shared Preferences
- **'com.squareup.picasso:picasso:2.71828'**
- Για την παρουσίαση της εικόνας προφίλ.

## 2. Διαγράμματα

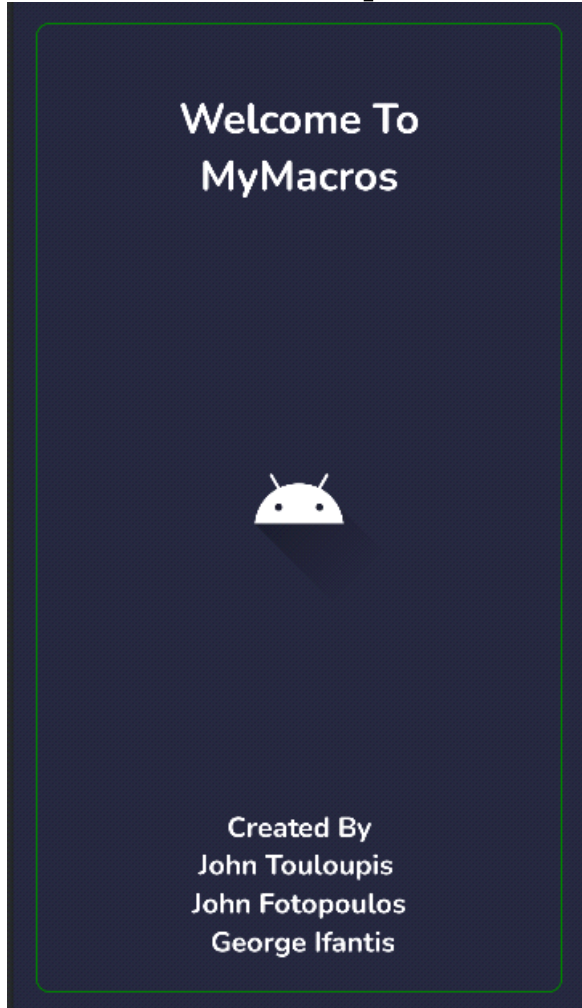
### Διάγραμμα Τάξεως

Δείτε UML-CLASS\_DIAGRAM.png

### Διάγραμμα Περιπτώσεων Χρηστών


Δείτε Use case diagram.png



### 3. Ανάλυση UI



Η σελίδα αυτή εμφανίζεται όταν ανοίγει για πρώτη φορά η εφαρμογή και εμφανίζεται μέχρι να αναγνωρίσει η εφαρμογή αν ο χρήστης είναι συνδεδεμένος ή όχι. Ανάλογα αν είναι ή όχι θα προχωρήσει στην σελίδα Login ή στο Main Activity – Diary Fragment (ή στο Quiz Activity αν δεν είναι verified).

# Sign in



 Email

 Password 

[Forgot Password](#)

Log in

or Sign in  
with

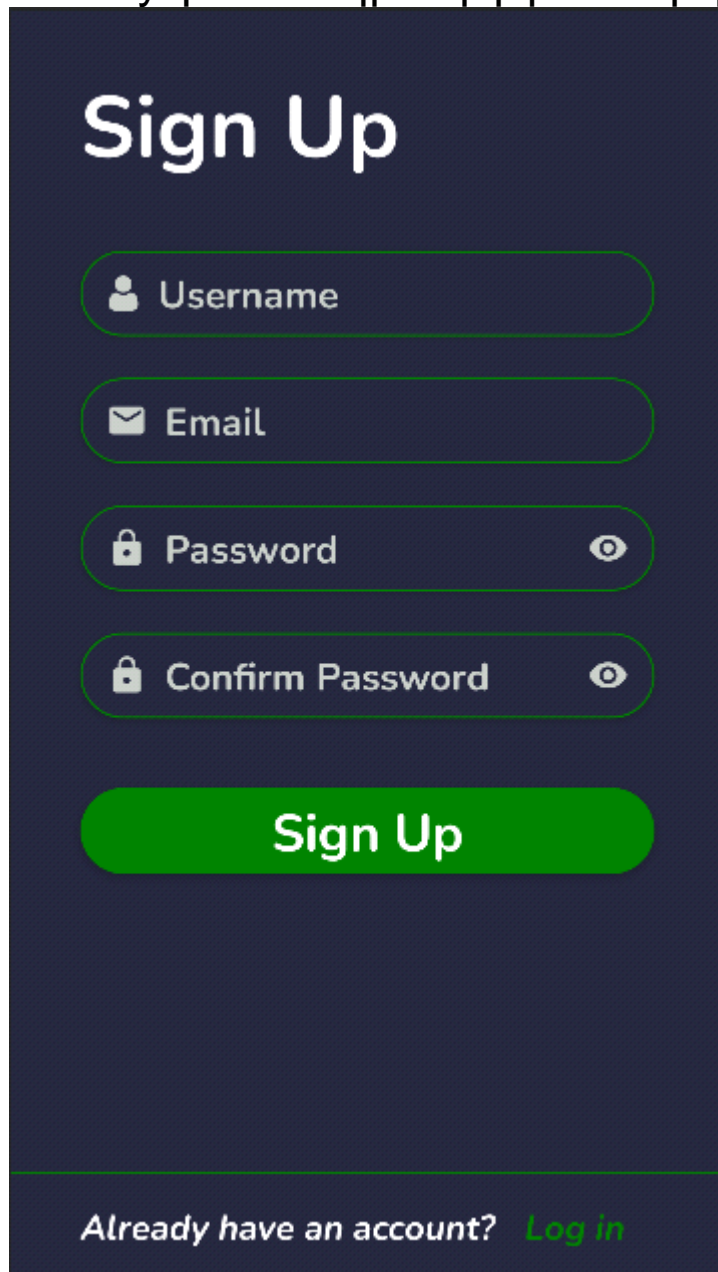
Dont have an account? [Sign Up](#)

Σε αυτή την σελίδα ο χρήστης μπορεί να συνδεθεί βάζοντας το email και τον κωδικό του, η με τον λογαριασμό Google η Twitter. Στην δεύτερη περίπτωση αν δεν υπάρχει το προφίλ θα δημιουργήσει καινούργιο αυτόματα, στο οποίο θα μπορείς να συνδεθείς μόνο με τον αντίστοιχο λογαριασμό.

Επιπλέον ο χρήστης δεν έχει λογαριασμό μπορεί να πατήσει το “Sign Up” για να πάει στην παρακάτω Register



Activity για να δημιουργήσει λογαριασμό.

A dark blue sign-up form with white text and icons. It features four input fields: Username (with a person icon), Email (with an envelope icon), Password (with a lock icon and a toggle eye icon), and Confirm Password (with a lock icon and a toggle eye icon). Below the fields is a large green button with the text "Sign Up". At the bottom, there is a link that says "Already have an account? Log in" where "Log in" is in green.

# Sign Up

Already have an account? [Log in](#)

Εδώ ο χρήστης μπορεί να δημιουργήσει λογαριασμό βάζοντας ένα όνομα χρήστη, email και κωδικό ασφαλείας. Επιπλέον μπορεί να πάει πίσω στην Login Activity πατώντας το κουμπί “Log in”. Αν φτιάξει νέο λογαριασμό θα προχωρήσει στην Quiz Activity για να δημιουργήσει το προφίλ του.

## Welcome Georgeifa

MyMacros will help you keep track of your calories, food habits and macros! With the help of this app you will be able to note down your meals, know how many proteins and other macros you need to consume in a day and many more!

Please fill up the quiz below, so we can help you more:

☒☐☐☒☐☐☒☐

Σε αυτό το Activity ο χρήστης πρέπει να δημιουργήσει το προφίλ του. Αν το δημιουργήσει θα θεωρηθεί verified και θα προχωρήσει το main activity. Αν δεν δημιουργήσει τότε θα θεωρείτε not verified και μέχρι να συμπληρώσει την σελίδα δεν θα μπορεί να προχωρήσει στην main activity.



TODAY



0  
EATEN

1928  
KCAL LEFT

0  
BURNT

CARBS

PROTEIN

FAT

0 / 193 gr

0 / 145 gr

0 / 64 gr



Add Breakfast

Recommended: 236 - 386 kcal



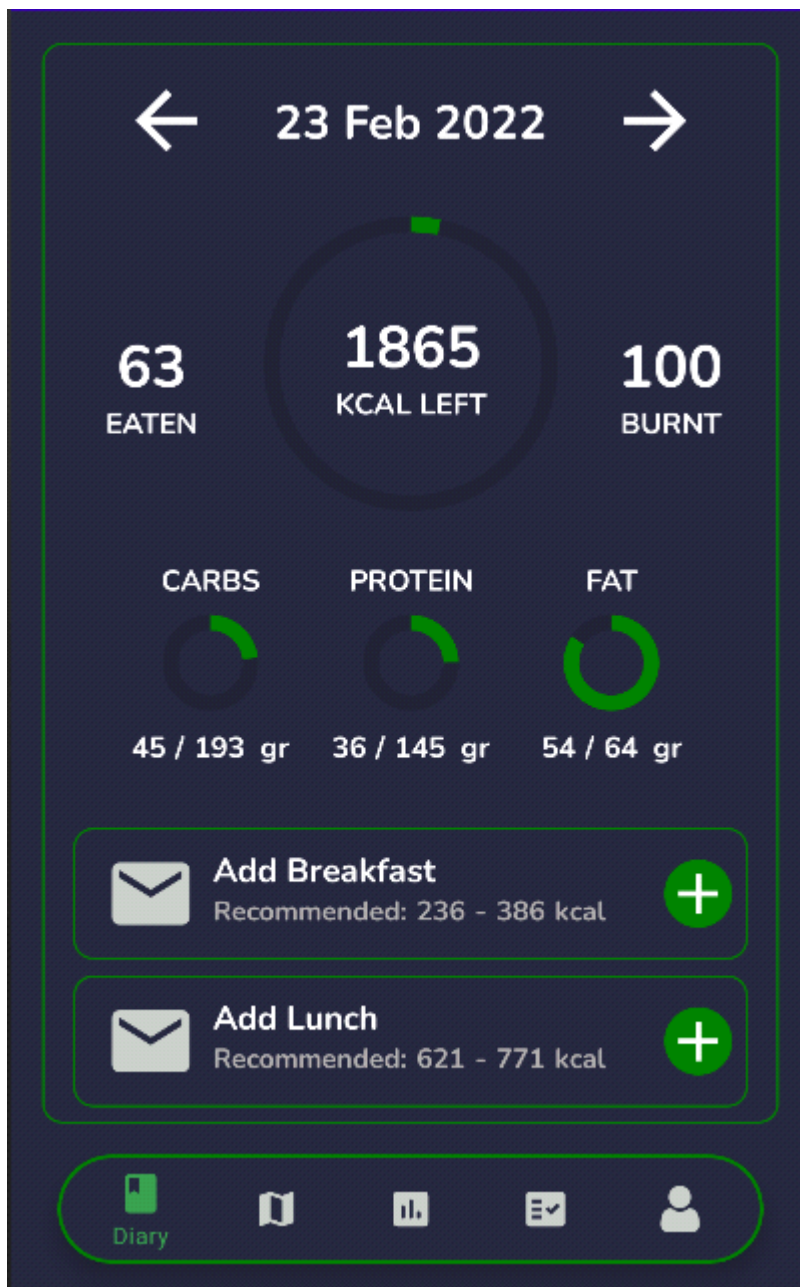
Add Lunch

Recommended: 621 - 771 kcal



Diary

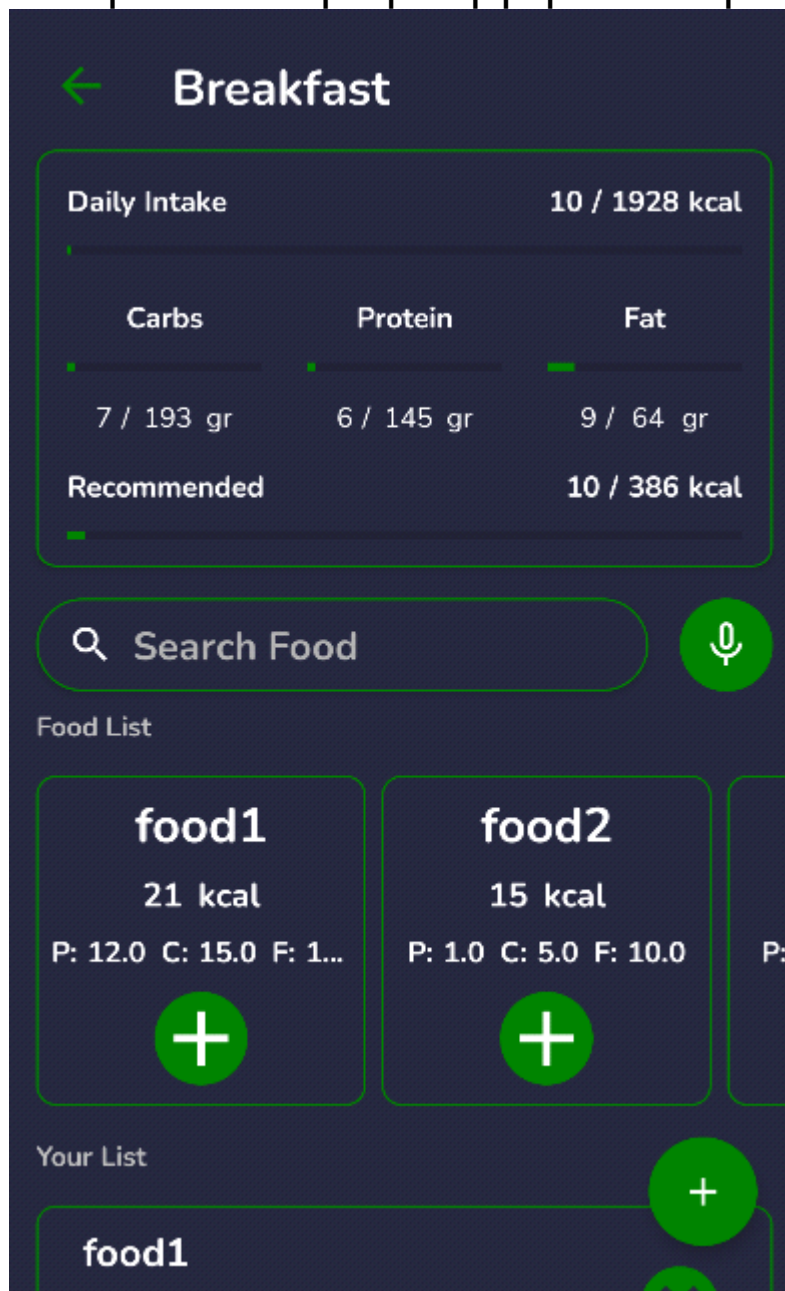


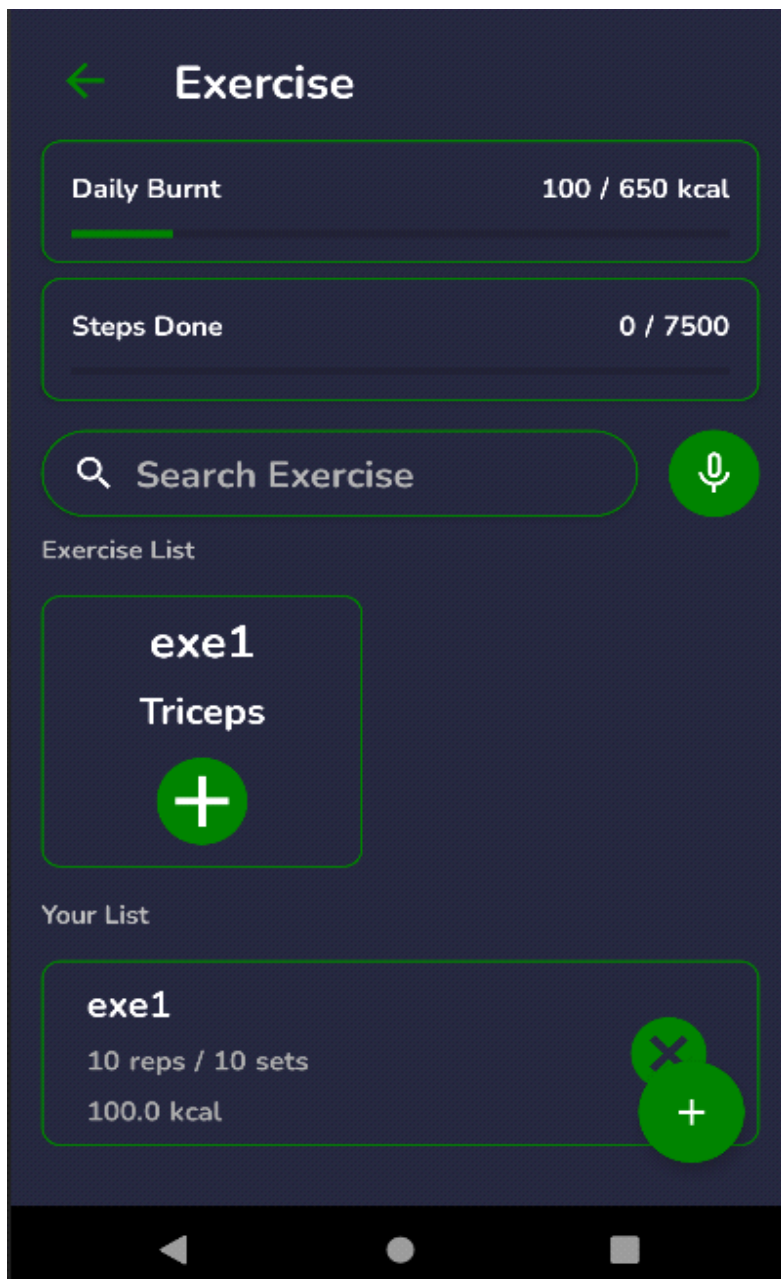


Στην σελίδα αυτή ο χρήστης μπορεί να δει πόσες θερμίδες έχει καταναλώσει, κάψει καθώς και πόσες του έχουν μείνει για την ημέρα αλλά και το πλήθος των μακροθρεπτικών συστατικών. Επιπλέον μπορεί με τα βελάκια πάνω να προχωρήσει σε προηγούμενες και επόμενες ημερομηνίες ή πατώντας στην ημερομηνία να επιστρέψουν στην σημερινή.

Επιπλέον πατώντας σε κάποιο από τα  στο μενού μπορεί

να προσθέσει γεύμα ή γυμναστική.





Στα δύο παραπάνω activities ο χρήστης μπορεί να δει το αντίστοιχο γεύμα (η την γυμναστική) της ημέρας που έχει επιλέξει, καθώς να προσθέσει ή να αφαιρέσει φαγητά σε αυτό το γεύμα (ή ασκήσεις στην γυμναστική). Επιπλέον ο χρήστης πατώντας το κουμπί κάτω δεξιά μπορεί να προσθέσει τα δικά του φαγητά (και ασκήσεις) τα οποία αποθηκεύονται τοπικά στην συσκευή του. Τέλος ο χρήστης μπορεί να αναζητήσει όποιο φαγητό (ή άσκηση θέλει) από

την λίστα με όλα τα φαγητά (και ασκήσεις), και τοπικά και απο την βάση). Για αυτή την λειτουργία μπορεί να χρησιμοποιήσει και το μικρόφωνο πατώντας το κουμπί με το μικρόφωνο.

Όταν ο χρήστης πατήσει για να προσθέσει ένα φαγητό στο γεύμα του ή άσκηση στο πλάνο γυμναστικής του βγαίνουν τα εξής dialog.

The image shows a dark-themed dialog box titled "food1". At the top, there is a rounded rectangular input field containing the number "50". Below this field, the following nutritional information is listed in a vertical stack:

- Calories: 10.5
- Protein: 6.0
- Carbs: 7.5
- Fat: 9.0

At the bottom of the dialog, there are two rounded buttons: a green "Add" button on the left and a red "Cancel" button on the right.

The image shows a dark-themed dialog box titled "exe1". It contains two input fields at the top, both containing the number "10". Below these is a label "Category" followed by the text "Triceps". Underneath is a label "Calories Burnt" followed by the value "100.0". At the bottom, there are two buttons: a green "Add" button and a red "Cancel" button.

Όταν ο χρήστης πατήσει να προσθέσει ένα δικό του φαγητό ή άσκηση εμφανίζεται το αντίστοιχο dialog.



## Add New Food

Food Title

0 / 50

Protein

Carbs

Fat

Calories per 100g

Add

Cancel

## Add New Exercise

Exercise Title

0 / 50

Biceps

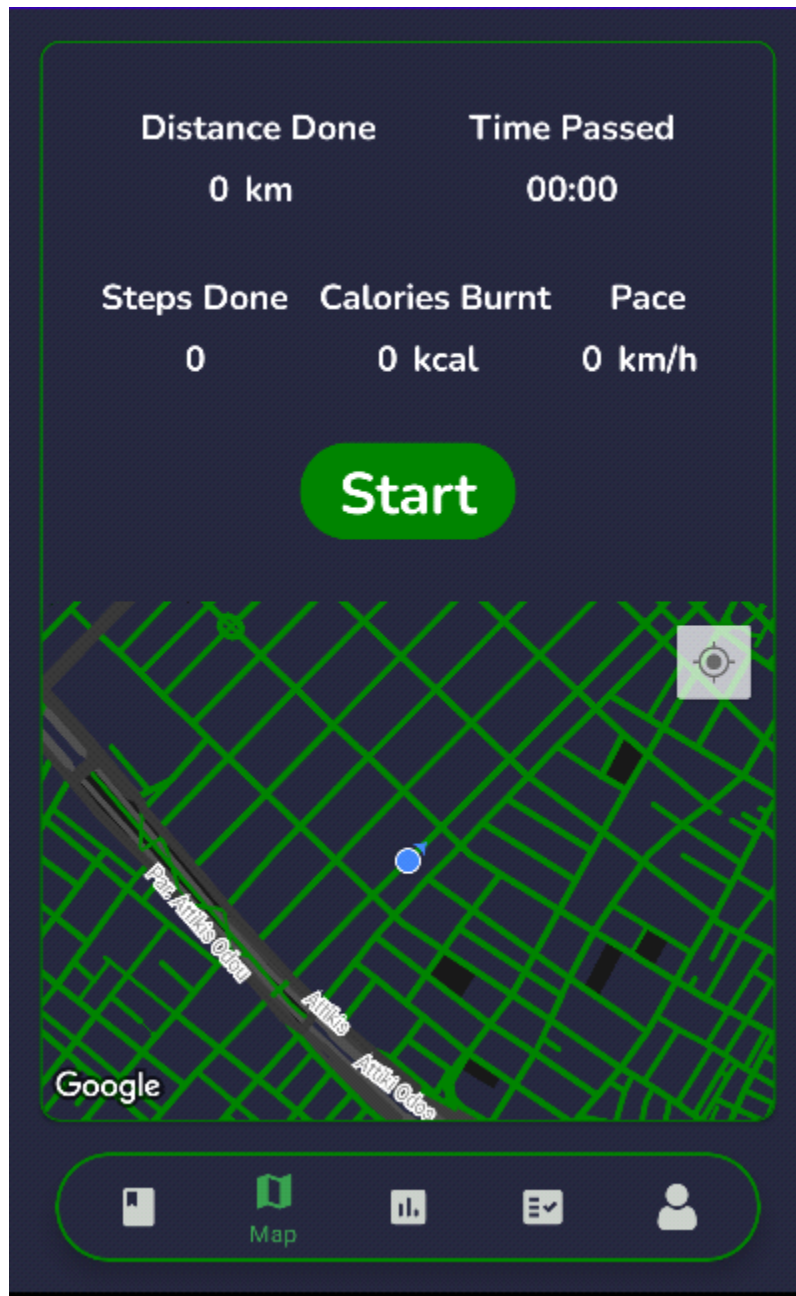


Calories Per 10 Reps

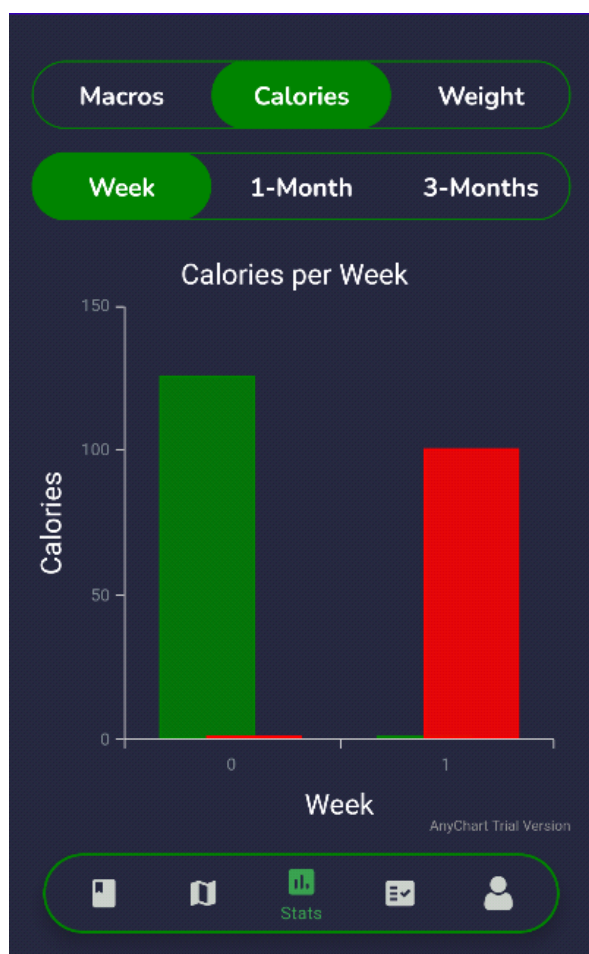
Add

Cancel

Ο χρήστης θα πρέπει να συμπληρώσει τα αντίστοιχα στοιχεία και όταν πατήσει “ADD” θα αποθηκευτεί τοπικά το φαγητό ή η άσκηση αντίστοιχα.



Εδώ ο χρήστης θα μπορεί να μετρήσει τα βήματα του, καθώς και τις θερμίδες, απόσταση και χρόνο που περπατάει. Επιπλέον στον χάρτη θα μπορεί να δει το που βρίσκεται καθώς και την διαδρομή που θα έχει κάνει όταν ήταν ενεργή η διαδικασία. Η διαδικασία όπως αναφέρθηκε και προηγούμενες είναι ενεργή απο όταν ο χρήστης πατήσει το “Start” μέχρι να πατήσει “Stop” η να βγει απο το fragment.



Macros

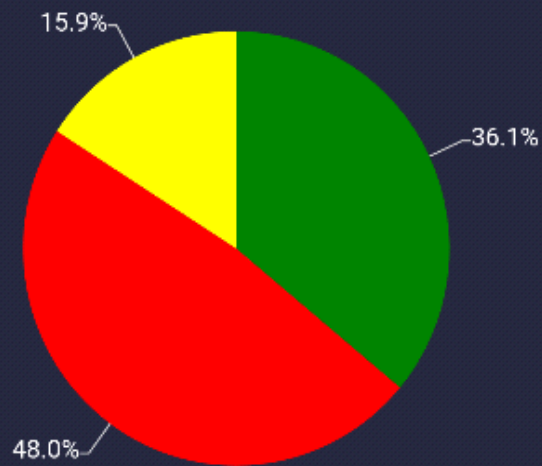
Calories

Weight

## Macronutrient Ratio

Macros

PROTEIN CARBS FAT



AnyChart Trial Version

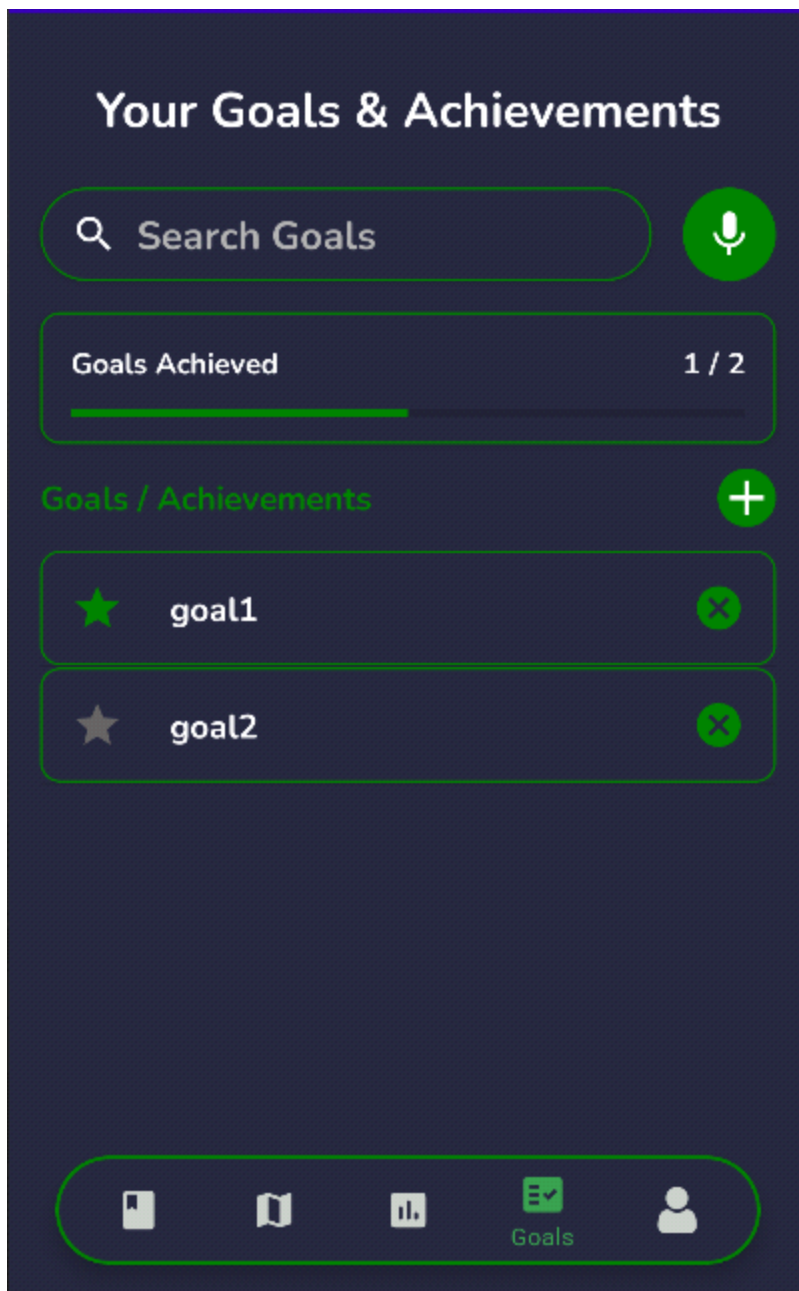


Stats





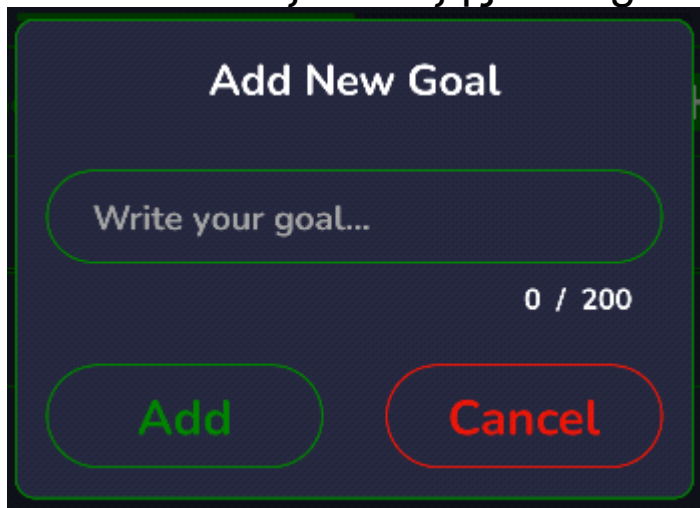
Στις παραπάνω εικόνες βλέπουμε το Stats Fragment εδώ ο χρήστης θα μπορεί να δει διάφορα στατιστικά όπως την κατανομή των μακροθρεπτικών συστατικών, την πορεία βάρους κλπ. Πέρα απο τα διαφορετικά αντικείμενα των στατιστικών για κάποια απο αυτά μπορεί να αλλάξει και την χρονική περίοδο.



Σε αυτή την σελίδα ο χρήστης θα μπορεί να προσθέσει, αφαιρέσει στόχους. Επιπλέον θα μπορείς να αλλάξει την κατάσταση τους αν τους έχει καταφέρει. Επιπλέον θα μπορεί να αναζητήσει οποιόν στόχο θέλει χρησιμοποιώντας την γραμμή αναζήτησης ή και το μικρόφωνο.

Για να προσθέσει στόχο θα πρέπει να πατήσει το + και

αυτό θα ανοίξει το εξής dialog.

A dark blue dialog box titled "Add New Goal". It features a text input field with the placeholder text "Write your goal..." and a character count "0 / 200" to its right. At the bottom, there are two buttons: "Add" in green and "Cancel" in red.

**Add New Goal**

Write your goal...

0 / 200

**Add** **Cancel**

Οι στόχοι – κατορθώματα αποθηκεύονται στην βάση δεδομένων.



t2  
t 2  
t2@gmail.com



## Account Details



t2



t2@gmail.com



Password



## Personal Details

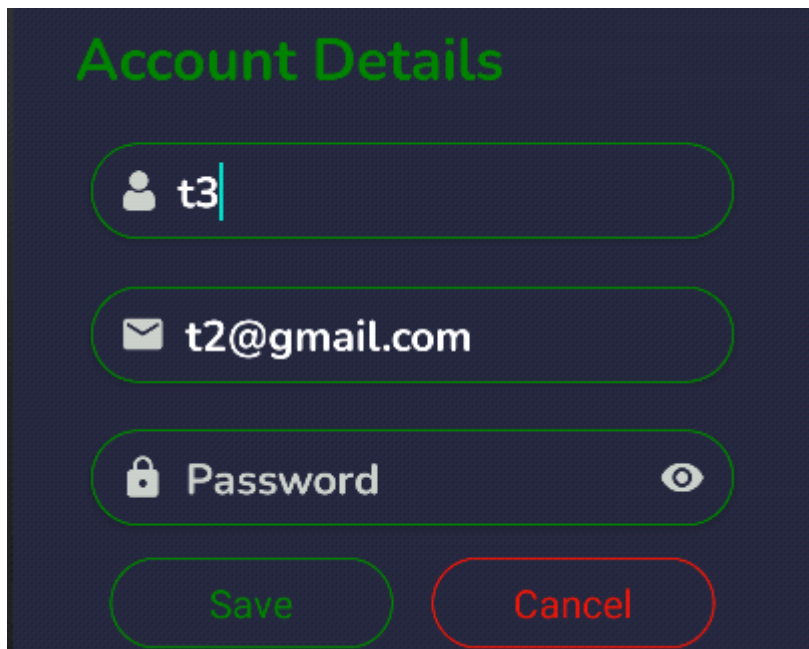
t

2





Account





A screenshot of a mobile application's 'Account Details' screen. The title 'Account Details' is at the top in green. Below it are three input fields: the first contains a person icon and 't3', the second contains an envelope icon and 't2@gmail.com', and the third contains a lock icon, the text 'Password', and an eye icon. At the bottom are two buttons: 'Save' in green and 'Cancel' in red.

## Account Details

 t3

 t2@gmail.com

 Password 

[Save](#) [Cancel](#)

Εδώ ο χρήστης μπορεί να δει τα στοιχεία που έχει προηγουμένως θέσει (απο το register και το Quiz). Επιπλέον μπορεί να αλλάξει οποιοδήποτε απο αυτά θέλει (για τον κωδικό και το email πρέπει να έχει φτιάξει κανονικό λογαριασμό και όχι third-party). Επιπλέον μπορεί να επιλέξει και να αλλάξει την εικόνα προφίλ του.



Τέλος μπορεί να πάει στο settings activity.

Αν πατήσει στην εικόνα προφίλ του θα τον πάει στο Gallery για να επιλέξει μια εικόνα.

Αν αλλάξει το email η το password του θα πρέπει να κάνει re-authenticate που θα γίνει με το εξής dialog.

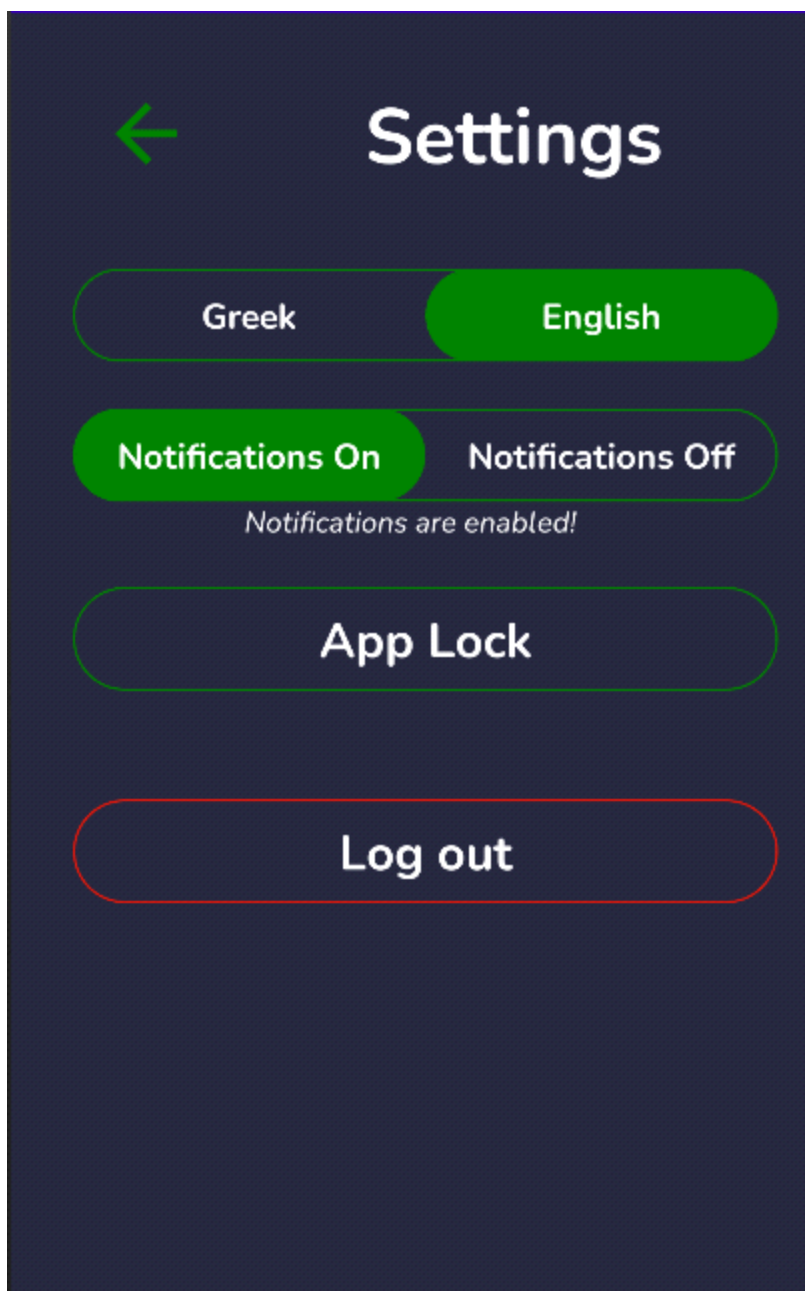
**Re-Authentication Required**

Please enter your password  
below or press the button  
from fingertip authentication:

 Password 

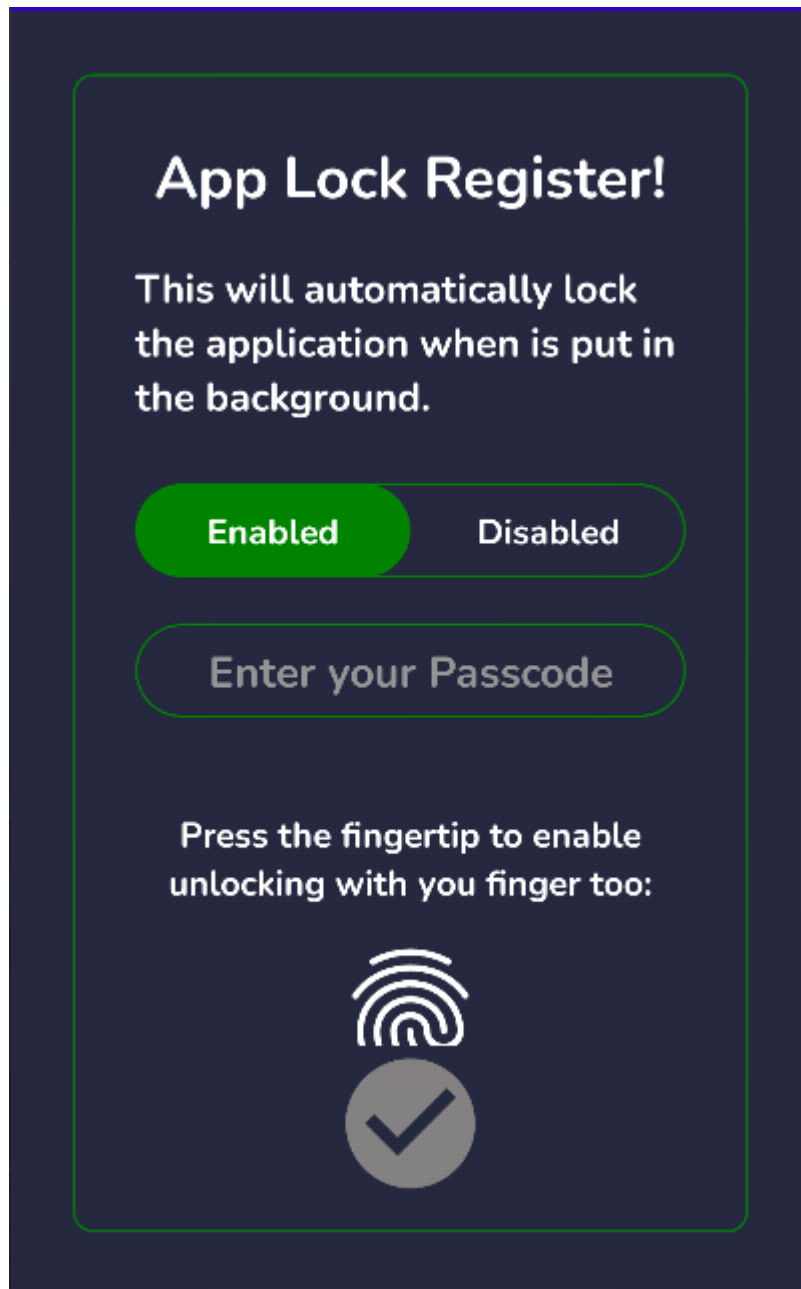
**Confirm** **Cancel**

Ο χρήστης θα πρέπει να ξαναβάλει τον κωδικό του ώστε να γίνει authenticated.



Εδώ ο χρήστης μπορεί να αλλάξει τις ρυθμίσεις του. Οι ρυθμίσεις αποθηκεύονται τοπικά στην συσκευή. Οι ρυθμίσεις που μπορεί να αλλάξει είναι η γλώσσα (μεταξύ ελληνικά και αγγλικά) όπου την αλλάζει σε ολόκληρη την εφαρμογή ακόμα και στις ειδοποιήσεις, να ενεργοποιήσει – απενεργοποιήσει τις ειδοποιήσεις και το App Lock που θα αναλυθεί παρακάτω. Επιπλέον από εδώ μπορεί να

αποσυνδεθεί.



Αυτή είναι η σελίδα που ο χρήστης μπορεί να κάνει register στο App Lock όταν το ενεργοποιεί. Υποχρεωτικά πρέπει έναν 4-ψήφιο κωδικό και έχει την επιλογή να ενεργοποιήσει το ξεκλείδωμα και με δαχτυλικό αποτύπωμα αν πατήσει το εικονίδιο. (πράσινο = ενεργό, άσπρο,γκρι = μη ενεργό).

Welcome Back!

Unlock with your fingerprint

Press to verify with Touch ID.

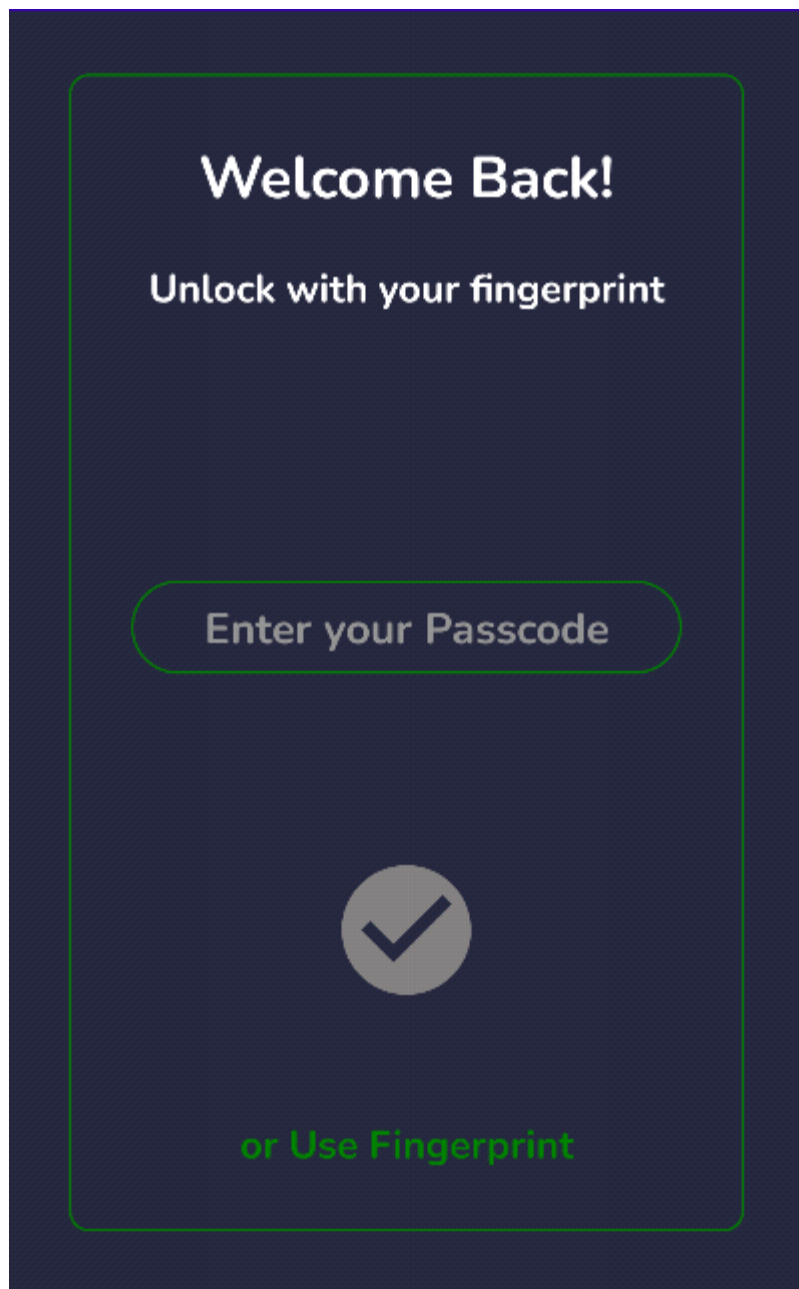


Verify with your fingerprint:



Touch the fingerprint sensor

[Use passcode](#)



Αυτές οι δύο εικόνες αποτελούν το App Lock το οποίο θα εμποδίζει τον χρήστη να μπει στην εφαρμογή αν δεν κάνει την πιστοποίηση. Το App Lock ενεργοποιείτε αυτόματα όταν η εφαρμογή είναι για πάνω απο 10 δευτερόλεπτα στο παρασκήνιο.

## 4. Ανάλυση Κώδικα

Όλα τα αρχεία με τον κώδικα περιέχουν σχόλια.  
Επομένως δεν θα αναλυθεί ολόκληρος ο κώδικας  
αλλά μόνο κάποια πιο ιδιαίτερα σημεία.

## ***Launch Activity***

Πραγματοποιείται έλεγχος κατά το άνοιγμα της εφαρμογής για το αν υπάρχει ήδη συνδεδεμένος χρήστης στην εφαρμογή για την συγκεκριμένη συσκευή. Αν υπάρχει συνδεδεμένος χρήστης που έχει ολοκληρώσει το quiz μεταφέρεται στο Main activity, σε αντίθετη περίπτωση μεταφέρεται στο Quiz activity.

```
if (myApp.getUser() == null){
    new Handler().postDelayed() -> {
        Intent intent = new Intent( packageContext, LoginActivity.class);

        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
        intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
        startActivity(intent);
    }, DelayMillis: 1000);
}else{
    myApp.setSettingsHelper();

    myApp.getdRef().child("User-Details")
        .child(myApp.getUser().getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                /*check if the user is verified (has completed the quiz)
                go to the main Activity
                otherwise to the Quiz Activity
                */

                UserProfile userP = snapshot.getValue(UserProfile.class);

                myApp.setUserProfile(userP);

                Intent intent;
                if(myApp.getUserProfile() != null){
                    if(myApp.getUserProfile().isVerified())
                        intent = new Intent( packageContext, MainActivity.class);
                    else
                        intent = new Intent( packageContext, LauncherActivity.this, QuizActivity.class);
                }else{
                    myApp.setUserProfile(new UserProfile( isVerified: false, userName: "", firstName: "", lastName: "", gender: "", age: 0, weight: 0, height: 0, goal: "", exFreq: ""));
                    intent = new Intent( packageContext, LauncherActivity.this, QuizActivity.class);
                }

                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
                intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                startActivity(intent);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });
};
```

## Login Activity

Αρχικά, ο χρήστης καλείται να συνδεθεί με έναν από τους τρεις διαθέσιμους διαφορετικούς τρόπους, α) email και κωδικό , β) Twitter sign in και γ) Google sign in.

Αφού επιτύχει η σύνδεση γίνεται επίσης έλεγχος για το αν ο χρήστης έχει ολοκληρώσει το Quiz activity κατευθύνοντας τον χρήστη στο κατάλληλο Activity.

```
private void goToNextActivity() {
    //Upon Login Check if user is verified (Completed task)
    //if he is go to main activity
    //if not go to Quiz Activity

    Application myApp = ((Application) getApplicationContext());

    myApp.setUser();

    dRef.child(myApp.getUser().getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            UserProfile userP = snapshot.getValue(UserProfile.class);

            myApp.setSettingsHelper();
            myApp.setUserProfile(userP);

            Intent intent;
            if(myApp.getUserProfile() != null){
                if(myApp.getUserProfile().isVerified()){
                    intent = new Intent( packageContext LoginActivity.this, MainActivity.class);
                }else{
                    intent = new Intent( packageContext LoginActivity.this, QuizActivity.class);
                }else{
                    myApp.setUserProfile(new UserProfile( isVerified: false, userName: "", firstName: "", lastName: "", gender: "", age: 0, weight: 0, height: 0, goal: "", excFreq: ""));
                    intent = new Intent( packageContext LoginActivity.this, QuizActivity.class);
                }
            }
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
            startActivity(intent);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}
```

## Register Activity



Αφού ο χρήστης συμπληρώσει τα στοιχεία του, γίνεται αποθήκευση στην βάση και εμφανίζεται μήνυμα επιτυχίας, ύστερα στο `createUserProfile()` δημιουργείται ένα προσωρινό User Profile μέχρι να το συμπληρώσει ο χρήστης στο Quiz Activity.

```
private void createUser(){
    if(inputsAreValid()){
        mAuth.createUserWithEmailAndPassword(email.getText().toString(),password.getText().toString()).addOnCompleteListener(task -> {
            if(task.isSuccessful()){
                Toast.makeText( context: RegisterActivity.this, text: "User Registered Successfully!", Toast.LENGTH_SHORT).show();
                createUserProfile();
            }else{
                Toast.makeText( context: RegisterActivity.this, text: "Registration Error:" + Objects.requireNonNull(task.getException()).getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}

//Create New User Profile And Go To Next Activity
private void createUserProfile() {
    UserProfile newUser = new UserProfile( isVerified: false, name.getText().toString(), firstName: "", lastName: "", gender: "", age: 0, weight: 0, height: 0, goal: "", excFreq: "");

    setGlobalValues(newUser);
    dRef.child(Objects.requireNonNull(mAuth.getCurrentUser()).getUid()).setValue(newUser);

    Intent intent =new Intent( packageContext: RegisterActivity.this, QuizActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
    intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);

    startActivity(intent);
}
```

## ***Quiz Activity***

Στο συγκεκριμένο σημείο του κώδικα ορίζονται στο User Profile τα απαιτούμενα στοιχεία και

αποθηκεύονται στο τέλος στην βάση.

```
//set the new user profile
private void setUserProfile() {
    userP.setFirstName(fName.getText().toString());
    userP.setLastName(lName.getText().toString());
    userP.setAge(Integer.parseInt(age.getText().toString()));
    userP.setWeight(Float.parseFloat(weight.getText().toString()));
    userP.setHeight(Float.parseFloat(height.getText().toString()));

    RadioButton rb = findViewById(genderRG.getCheckedRadioButtonId());
    userP.setGender(rb.getTag().toString());

    rb = findViewById(planRG.getCheckedRadioButtonId());
    String tmp = "";
    switch (rb.getId()){
        case R.id.lose_quiz_SWT:
            tmp = "lose";
            break;
        case R.id.maintain_quiz_SWT:
            tmp = "maintain";
            break;
        case R.id.gain_quiz_SWT:
            tmp = "gain";
            break;
    }
    userP.setGoal(tmp);

    rb = findViewById(exeRG.getCheckedRadioButtonId());
    switch (rb.getId()){
        case R.id.noExe_quiz_SWT:
            tmp = "low";
            break;
        case R.id.moderateExe_quiz_SWT:
            tmp = "moderate";
            break;
        case R.id.highExe_quiz_SWT:
            tmp = "high";
            break;
    }

    userP.setExcFreq(tmp);

    userP.setVerified(true);

    myApp.setUserProfile(userP);

    dRef.child(mUser.getUid()).setValue(userP);

    startActivity(new Intent( packageContext: this, MainActivity.class));
}
}
```

***Diary Fragment***

Συνάρτηση initializePB(). Οριοθετούμε με minimum την τιμή 0 τις 4 βασικές μεταβλητές kcalPB, protPB, carbPB, fatPB τύπου ProgressBar. Εντός της onDataChange αρχικά ελέγχουμε αν η επιλογή του χρήστη είναι “Exercise” ή “Food”. Στην πρώτη περίπτωση με την κάνουμε έναν δεύτερο έλεγχο για το ποιο από τα παιδιά έχει επιλέξει είτε “Steps” είτε “Exercise”. Και στις δύο περιπτώσεις εκχωρούμε στην μεταβλητή s\_c την τιμή μέσω των βημάτων ή την εκτέλεση των ασκήσεων μέσω των αντίστοιχων getter από την class Step\_Calories.

Αν η επιλογή ήταν “food” εκχωρούμε στην μεταβλητή uF την τιμή των θερμίδων μέσω getter από την class User\_Food. Στη συνέχεια προσθέτουμε (μαζί με την προηγούμενη τιμή) στην μεταβλητή eatenC τις θερμίδες κάθε φαγητού.

```

150
151 private void initializePB() {
152     kcalPB.setMax(recomm_c);
153     kcalPB.setMin(0);
154
155     protPB.setMax(recomm_prot);
156     protPB.setMin(0);
157     fatPB.setMax(recomm_fat);
158     fatPB.setMin(0);
159     carbPB.setMax(recomm_carb);
160     carbPB.setMin(0);
161
162     fat_recTXT.setText(String.valueOf(recomm_fat));
163     prot_recTXT.setText(String.valueOf(recomm_prot));
164     carbs_recTXT.setText(String.valueOf(recomm_carb));
165
166
167     DateFormat df = new SimpleDateFormat("yyyy-MM-dd", Locale.ENGLISH);
168     dRef.child("User_Record").child(mUser.getId()).child(df.format(date)).addListenerForSingleValueEvent(new ValueEventListener() {
169         @Override
170         public void onDataChange(@NonNull DataSnapshot snapshot) {
171             int eatenC= 0;
172             int burntC= 0;
173             int prot=0;
174             int fat=0;
175             int carb = 0;
176             for(DataSnapshot snap:snapshot.getChildren()){
177                 if(Objects.equals(snap.getKey(), "Exercise")){
178                     for(DataSnapshot s:snap.getChildren()){
179                         if(Objects.equals(s.getKey(), "Steps")){
180                             Step_Calories s_c = s.getValue(Step_Calories.class);
181                             assert s_c != null;
182                             burntC += s_c.getCalories();
183                         }else{
184                             User_Exercise uE = s.getValue(User_Exercise.class);
185                             assert uE != null;
186                             burntC += ((uE.getReps() * uE.getSets())*uE.getExe().getCalories_per_10_reps())/10;
187                         }
188                     }
189                 }else{
190                     for(DataSnapshot s:snap.getChildren()){
191                         User_Food uF = s.getValue(User_Food.class);
192                         assert uF != null;
193                         eatenC += (uF.getAmount() * uF.getFood().getCalories_per_100g())/100;
194
195                         prot+= (uF.getAmount() * uF.getFood().getProt_per_100g())/100;
196                         carb+= (uF.getAmount() * uF.getFood().getCarbs_per_100g())/100;
197                         fat+= (uF.getAmount() * uF.getFood().getFat_per_100g())/100;
198                     }
199                 }
200             }

```

## Map Fragment

Στο πάτημα του κουμπιού start, αρχικά γίνεται έλεγχος για το αν έχει δοθεί άδεια εντοπισμού (gps) για τον εντοπισμό του χρήστη στον χάρτη. Στην συνέχεια χρησιμοποιούνται οι μεταβλητές newRoute, firstStart και start για την "χάραξη" και έναρξη

καινούργιας πορείας και αριθμό βημάτων.

```
private void startClick(){
    if(ContextCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED){
        checkPermission();
    }else{
        newRoute=true;

        firstStart = true;
        start = true;

        startTimer();
    }
}
```

## ***Stats Fragment***

Η συγκεκριμένη συνάρτηση είναι υπεύθυνη για την επιλογή του κατάλληλου διαγράμματος, είτε κυκλικό διάγραμμα για την εμφάνιση και κατανομή των πρωτεϊνών, υδανθράκων και λιπαρών είτε διάγραμμα

στηλών για την εμφάνιση της προόδου στις θερμίδες και το βάρος.

```
private void chooseChartType(){
    chartRG.setOnCheckedChangeListener((group, checkedId) -> {
        if(checkedId == R.id.overview_stats_SWT){
            resetChartView();
            timeRG.setVisibility(View.GONE);

            createPieChart();
        }else{
            resetChartView();
            timeRG.setVisibility(View.VISIBLE);

            createColumnChart();
        }
    });

    //region Change Period
    timeRG.setOnCheckedChangeListener((group, checkedId) -> {
        resetChartView();

        createColumnChart();
    });
    //endregion
}
```

## ***Goals Fragment***

Στην class GoalsFragment εμπεριέχονται οι `onSomethingMethods()`, και `initializeSpeechRecognition()` ως βασικές συναρτήσεις. Στην πρώτη συνάρτηση υλοποιείται

ένα searchbar με το οποίο ο χρήστης έχει την δυνατότητα να

```
78 private void onSomethingMethods() {
79     addBTN.setOnClickListener(v -> {
80         AddGoalDialog cdd = new AddGoalDialog(context);
81         cdd.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
82         cdd.show();
83     });
84
85     //search for the foods that have the text in the search bar in their title (case sensitive)
86     search.addTextChangedListener(new TextWatcher() {
87         @Override
88         public void beforeTextChanged(CharSequence s, int start, int count, int after) {
89
90         }
91
92         @Override
93         public void onTextChanged(CharSequence s, int start, int before, int count) {
94         }
95
96         @Override
97         public void afterTextChanged(Editable s) {
98             //if search bar is empty load all the available foods
99             if(s.toString().isEmpty()){
100                 getGoals();
101             }else{
102                 ArrayList<Goal> searchGoals = new ArrayList<>();
103
104                 for(Goal g: goals){
105                     if(g.getDetails().toLowerCase(Resources.getSystem().getConfiguration().locale).contains(s.toString()))
106                     {
107                         searchGoals.add(g);
108                     }
109                 }
110
111                 adapter = new GoalsAdapter(context,searchGoals);
112                 listView.setAdapter(adapter);
113
114                 initializeListView();
115                 setVisibility(searchGoals);
116             }
117         }
118     });
119
120
121     mic.setOnClickListener(v -> {
122         // check for permissions to user mic
123         if(ContextCompat.checkSelfPermission(context, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED){
124             checkPermission();
125         }else{
126             initializeSpeechRecognition();
127         }
128     });
129 }
```

## ***Account Fragment***

Στο AccountFragment υλοποιούνται οι λειτουργίες επεξεργασίας και αποθήκευσης των δεδομένων του χρήστη. Η συνάρτηση saveProfileDetails δέχεται όρισμα τύπου View το οποίο για την χρήση της μεταβλητής rb που αφορά το radio button.



```

245
246 @SuppressWarnings("NonConstantResourceId")
247 @
248 private void saveProfileDetails(View v) {
249     userP.setFirstName(firstN.getText().toString());
250     userP.setLastName(lastN.getText().toString());
251     userP.setAge(Integer.parseInt(age.getText().toString()));
252     userP.setWeight(Float.parseFloat(weight.getText().toString()));
253     saveWeight();
254     userP.setHeight(Float.parseFloat(height.getText().toString()));
255
256     RadioButton rb = v.findViewById(gender.getCheckedRadioButtonId());
257     userP.setGender(rb.getTag().toString());
258
259     rb = v.findViewById(plan.getCheckedRadioButtonId());
260     String tmp = "";
261     switch (rb.getId()){
262         case R.id.lose_acc_SWT:
263             tmp = "lose";
264             break;
265         case R.id.maintain_acc_SWT:
266             tmp = "maintain";
267             break;
268         case R.id.gain_acc_SWT:
269             tmp = "gain";
270             break;
271     }
272     userP.setGoal(tmp);
273
274     rb = v.findViewById(exercise.getCheckedRadioButtonId());
275     switch (rb.getId()){
276         case R.id.noExe_acc_SWT:
277             tmp = "low";
278             break;
279         case R.id.moderateExe_acc_SWT:
280             tmp = "moderate";
281             break;
282         case R.id.highExe_acc_SWT:
283             tmp = "high";
284             break;
285     }
286
287     userP.setExcFreq(tmp);
288
289     userP.setVerified(true);
290
291     myApp.setUserProfile(userP);
292     dRef.child(mUser.getUid()).setValue(userP);
293
294     resetView();
295 }

```

## Settings Activity

Στην `onSomethingMethods()` χρησιμοποιούνται radio buttons για την επιλογή της επιθυμητής γλώσσας της εφαρμογής. Η αποθήκευση των ρυθμίσεων πραγματοποιείται στην συνάρτηση `saveSettings`. Για



τις 3 λειτουργίες (αλλαγή γλώσσας, χρήση ειδοποιήσεων, χρήση κλειδώματος εφαρμογής) χρησιμοποιούνται radio buttons όπως επίσης και on/off button, και ανάλογα την επιλογή του χρήστη γίνονται κατάλληλοι έλεγχοι.

```
78 private void onSomethingMethods() {
79     addBTN.setOnClickListener(v -> {
80         AddGoalDialog cdd = new AddGoalDialog(context);
81         cdd.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));
82         cdd.show();
83     });
84
85     //search for the foods that have the text in the search bar in their title (case sensitive)
86     search.addTextChangedListener(new TextWatcher() {
87         @Override
88         public void beforeTextChanged(CharSequence s, int start, int count, int after) {
89
90         }
91
92         @Override
93         public void onTextChanged(CharSequence s, int start, int before, int count) {
94         }
95
96         @Override
97         public void afterTextChanged(Editable s) {
98             //if search bar is empty load all the available foods
99             if(s.toString().isEmpty()){
100                 getGoals();
101             }else{
102                 ArrayList<Goal> searchGoals = new ArrayList<>();
103
104                 for(Goal g: goals){
105                     if(g.getDetails().toLowerCase(Resources.getSystem().getConfiguration().locale).contains(s.toString()))
106                     {
107                         searchGoals.add(g);
108                     }
109                 }
110
111                 adapter = new GoalsAdapter(context,searchGoals);
112                 listView.setAdapter(adapter);
113
114                 initializeListView();
115                 setVisibility(searchGoals);
116             }
117         }
118     });
119
120     mic.setOnClickListener(v -> {
121         // check for permissions to user mic
122         if(ContextCompat.checkSelfPermission(context, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED){
123             checkPermission();
124         }else{
125             initializeSpeechRecognition();
126         }
127     });
128 }
```

## ***Lock Register***

Αρχικά ελέγχουμε με την χρήση if αν ο χρήστης θέλει να χρησιμοποιήσει app lock. Με την μεταβλητή fingerprint τύπου ImageView υλοποιούμε το

## fingerprint scanner.

```
69 private void onSomethingMethods() {
70     enabledSWT.setOnCheckedChangeListener((group, checkedId) -> {
71         if(checkedId == R.id.enable_loc_REG_SWT){
72             passCode.setEnabled(true);
73             infoTXT.setTextColor(Color.WHITE);
74             fingerprint.setImageResource(R.drawable.ic_fingerprint);
75             okBTN.setImageResource(R.drawable.ic_check_disabled);
76             okBTN_state = false;
77             lockEnabled = true;
78         }else{
79             passCode.setEnabled(false);
80             passCode.clearFocus();
81             passCode.setText("");
82             infoTXT.setTextColor(Color.parseColor("colorString: "#919191"));
83             fingerprint.setImageResource(R.drawable.ic_fingerprint_disabled);
84             okBTN.setImageResource(R.drawable.ic_check_enabled);
85             okBTN_state = true;
86             lockEnabled = false;
87         }
88     });
89
90     //click on fingerprint to enable Unlocking with TouchId
91     fingerprint.setOnClickListener(v -> {
92         if(fingerPrint_state){
93             fingerprint.setImageResource(R.drawable.ic_fingerprint);
94             fingerPrint_state = false;
95         }else{
96             fingerprint.setImageResource(R.drawable.ic_fingerprint_enabled);
97             fingerPrint_state = true;
98         }
99     });
100 }
```

## ***Notification Receiver***

Στον παρακάτω κλωδικα δημιουργούμε ένα notification υπενθύμισης καταγραφής γευμάτων για τον χρήστη.

```

public class Notification_Receiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        NotificationManager notificationManager = (NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "Don't Forget";
            String description = "It's important to correctly add and track your meals an...";
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel( id: "MyMacros", name, importance);
            channel.setDescription(description);

            // Register the channel with the system; you can't change the importance
            // or other notification behaviors after this
            notificationManager = (NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
            notificationManager.createNotificationChannel(channel);
        }

        Intent repeating_intent = new Intent(context, MainActivity.class);

        repeating_intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
            Intent.FLAG_ACTIVITY_SINGLE_TOP);

        PendingIntent pendingIntent = PendingIntent.getActivity(context, requestCode: 100, repeating_intent, PendingIntent.FLAG_UPDATE_CURRENT);

        NotificationCompat.Builder builder = new NotificationCompat.Builder(context, channelid: "MyMacros")
            .setSmallIcon(R.drawable.ic_notifications)
            .setContentTitle(Html.fromHtml( source: "<b><font font-size:17px>" +
                "Don't Forget" +
                "</font><b>"))
            .setContentText(Html.fromHtml( source: "<i><font font-size:13px color=\\\"#FF757575\\\">" +
                "It's important to correctly add and track your meals an..." +
                "</font></i>"))
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
            .setAutoCancel(true);

        builder.setContentIntent(pendingIntent);

        notificationManager.notify( id: 100, builder.build());
    }
}

```

## ***Recommended Values Helper***

Στη συνάρτηση calculateRec() χρησιμοποιείται ο τύπος του bmi και σε συνδιασμό με τα στοιχεία και τους στόχους που έχει δώσει ο χρήστης εξάγεται ο αριθμός των θερμίδων και τα θρεπτικά συστατικά

που θα πρέπει να καταναλώσει μέσα στην μέρα.

```
public void calculateRec(UserProfile userP){
    switch (userP.getGender()){
        case "Male":
            bmr = 10*userP.getWeight()+6.25*userP.getHeight()-5*userP.getAge()+5;
            break;
        case "Female":
            bmr = 10*userP.getWeight()+6.25*userP.getHeight()-5*userP.getAge()+161;
            break;
    }

    switch (userP.getExcFreq()){
        case "low":
            excMultiplier = 1.2f;
            break;
        case "moderate":
            excMultiplier = 1.5f;
            break;
        case "high":
            excMultiplier = 1.7f;
            break;
    }

    bmr = bmr*excMultiplier;

    switch (userP.getGoal()){
        case "lose":
            bmr-=500;
            cal_toBurn += 150;
            break;
        case "gain":
            bmr+=300;
            cal_toBurn += 100;
            break;
    }

    prot = Math.round((float) ((0.3f*bmr)/4)); // each prot 4 calories
    carbs = Math.round((float) ((0.4f*bmr)/4)); //each carb 4 calories
    fat = Math.round((float) ((0.3f*bmr)/9)); //each carb 9 calories

    calories = (int) Math.round(bmr);
}
```

## ***Application***

Στην κλάση Application, χρησιμοποιείται ένα alarm service που ορίζει την συγκεκριμένη ώρα εμφάνισης notification για τον χρήστη.

```

//set the exact time the user receives a notification
public void initializeNotifications() {
    if(getSettings().isNotifications()) {
        Calendar calendar = Calendar.getInstance();

        calendar.set(Calendar.HOUR_OF_DAY, 12); //set hour
        calendar.set(Calendar.MINUTE, 0); //set minute
        calendar.set(Calendar.SECOND, 0); //set seconds
        Intent intent = new Intent(getApplicationContext(), Notification_Receiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(getApplicationContext(), requestCode 100, intent, PendingIntent.FLAG_UPDATE_CURRENT);

        AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
        alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), AlarmManager.INTERVAL_DAY, pendingIntent);
    }else{
        Intent intent = new Intent(getApplicationContext(), Notification_Receiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(getApplicationContext(), requestCode 100, intent, PendingIntent.FLAG_UPDATE_CURRENT);

        AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
        alarmManager.cancel(pendingIntent);
    }
}
}

```

Τέλος, χάρις το εικονιζόμενο int array, μπορούμε και παίρνουμε από διαφορετικές κλάσεις τα απαιτούμενα θρεπτικά συστατικά και θερμίδες ανάλογα από που καλείται.

```

//region Get The Recommended Macros
public int[] getRecommendedMacros(String specific){
    RecommendedValuesHelper helper;
    if(specific.isEmpty()) {
        helper = new RecommendedValuesHelper(getUserProfile());
        return new int[]{
            helper.getCalories(),
            helper.getProt(),
            helper.getCarbs(),
            helper.getFat(),
            helper.getCal_toBurn()
        };
    }
    else {
        helper = new RecommendedValuesHelper(getUserProfile(), specific);
        return new int[]{
            helper.getCalories(),
            helper.getProt(),
            helper.getCarbs(),
            helper.getFat(),
            helper.getSpecific_cal()
        };
    }
}
}

```