Food Locker®

Alessandro Giannetti
Edoardo Bini
Andrea Napoletani

Project for mobile applications and cloud computing

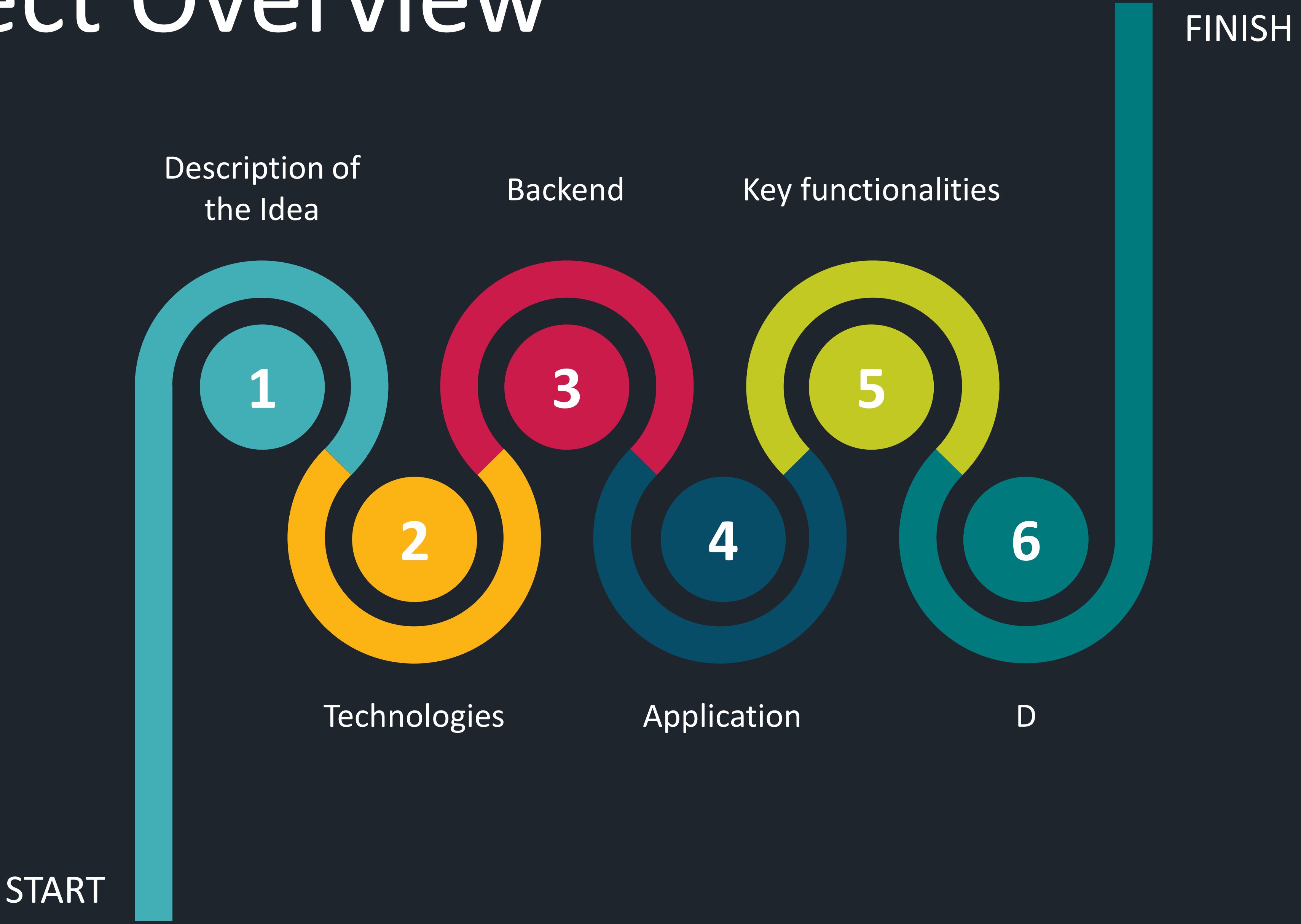
Our Team

Alessandro Giannetti

Andrea Napoletani

Edoardo Bini

Project Overview



Idea

1

CHAT

Keep in touch with your friends using the message system

2

PROGRESS

Keep track of your progress over time of weight loss and/or weight gain

3

DIARY

Keep track of calories and macronutrients from food eaten, and keep track of calories consumed with exercises

4

SOCIAL

Socialize with your friends through posts and share your achievements

Technologies

Backend

RAILS

Web-application framework for creation of database-backend

HEROKU

Container-based cloud Platform as a Service (PaaS)

Firebase

Backend-as-a-Service (BaaS)

Frontend

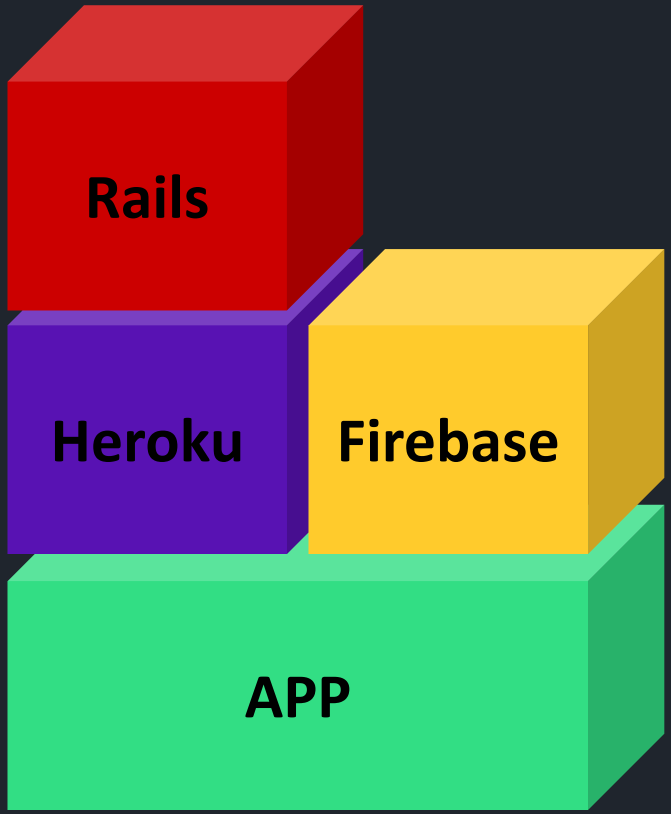
Kotlin

Open-source programming language that supports both object-oriented and functional programming

Backend Structure

with rails we have developed all those functions that relate to the **user's progress** the **diary** and the **social (posts and relations)**

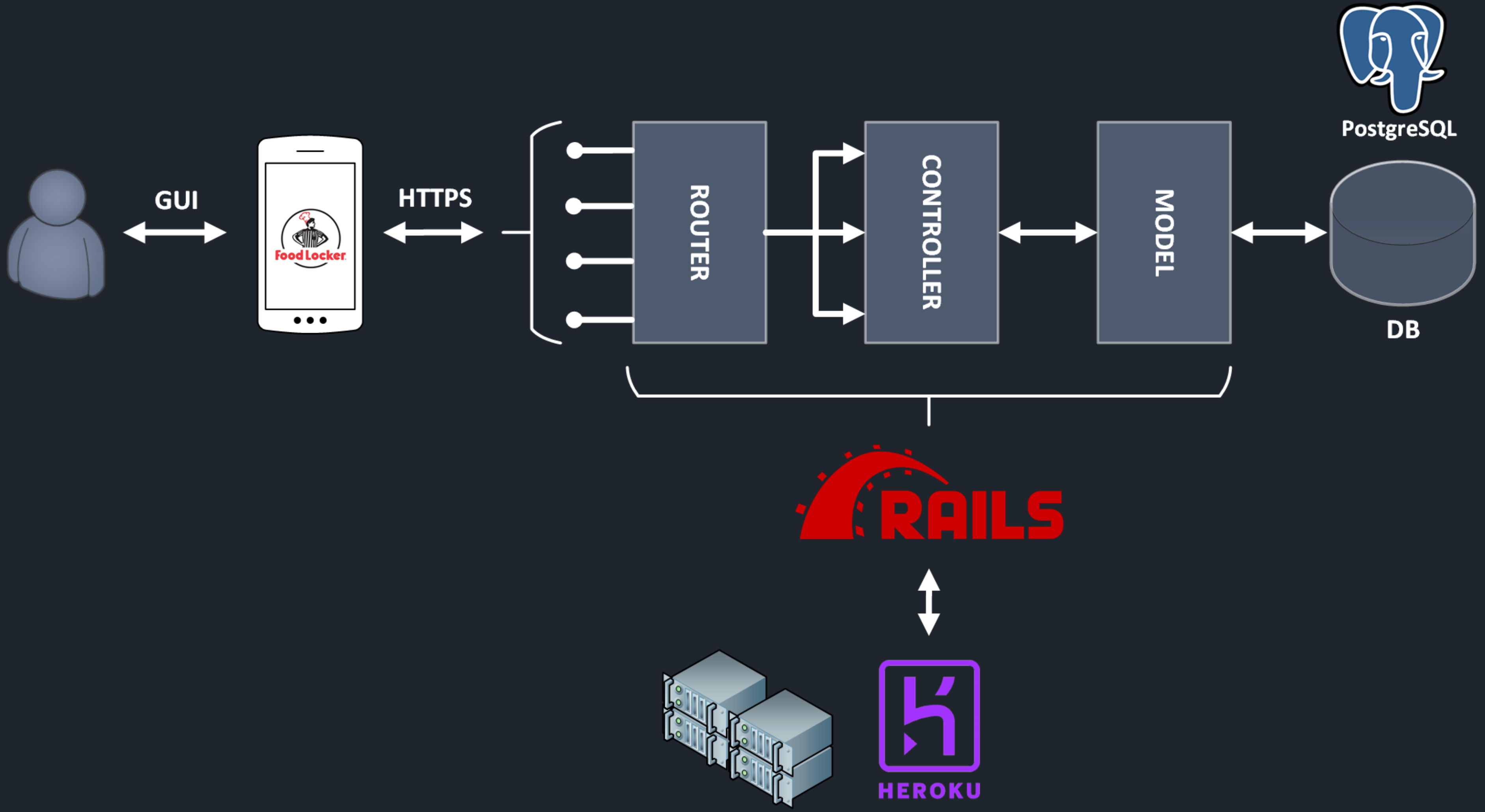
Hosting the rails server



Here we have managed the **chats**, **authentication features** and the **storage of images**

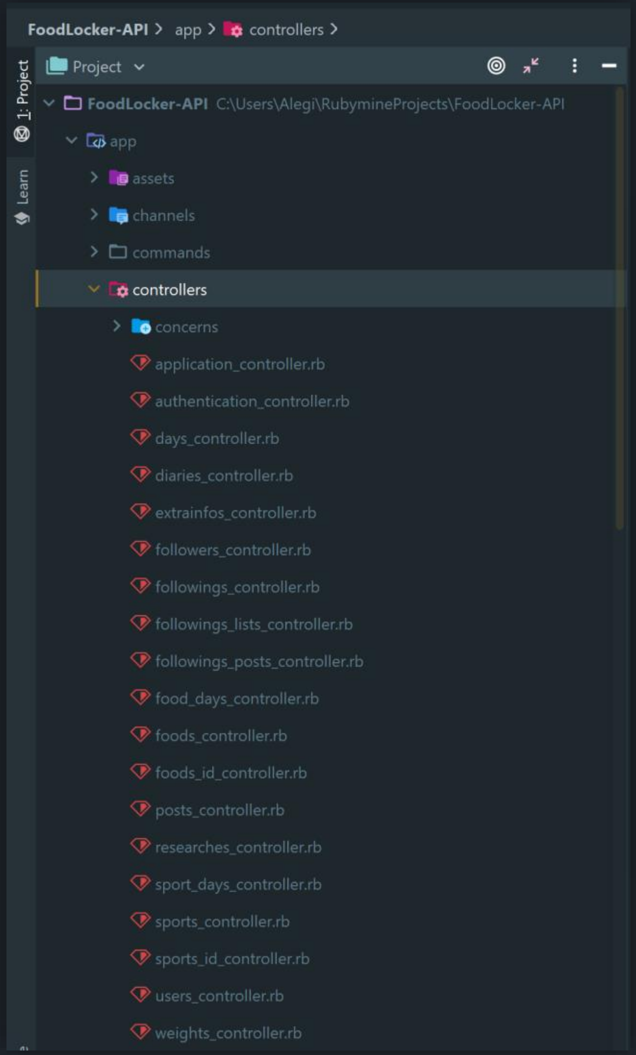
Android Application (MVVM pattern)

Rails + Heroku

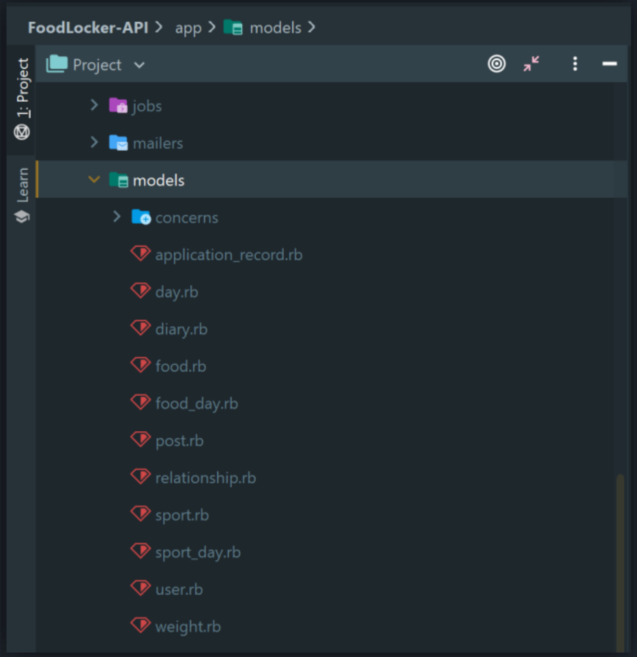


Rails Main Structure

This is the structure of the backend. Following the **MVC pattern**, here we have Models and Controllers, while the View is fully implemented on Android

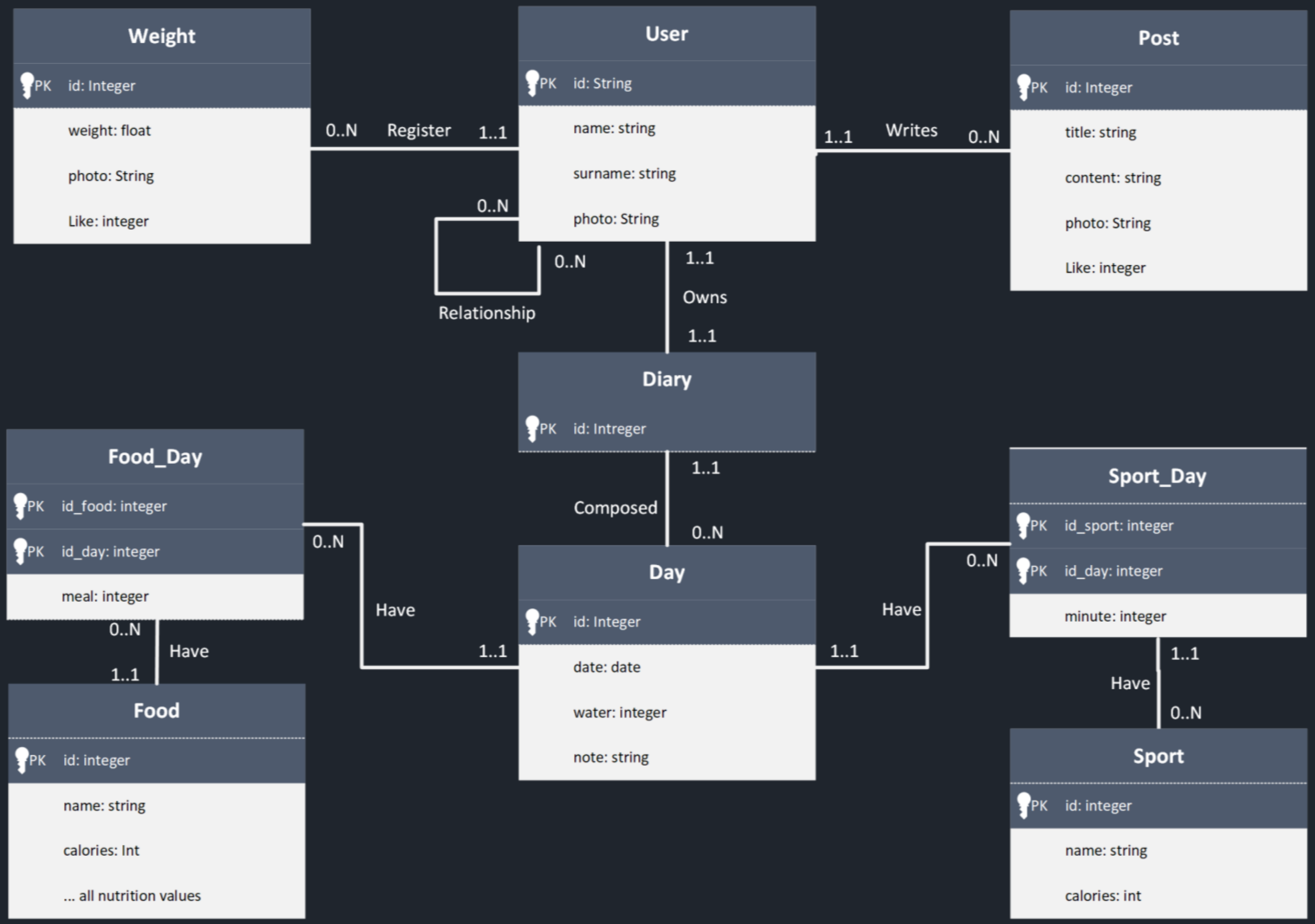


```
FoodLocker-API > app > controllers >
Project
FoodLocker-API  C:\Users\Alegi\RubymineProjects\FoodLocker-API
  app
    assets
    channels
    commands
    controllers
      concerns
      application_controller.rb
      authentication_controller.rb
      days_controller.rb
      diaries_controller.rb
      extrainfos_controller.rb
      followers_controller.rb
      followings_controller.rb
      followings_lists_controller.rb
      followings_posts_controller.rb
      food_days_controller.rb
      foods_controller.rb
      foods_id_controller.rb
      posts_controller.rb
      researches_controller.rb
      sport_days_controller.rb
      sports_controller.rb
      sports_id_controller.rb
      users_controller.rb
      weights_controller.rb
```

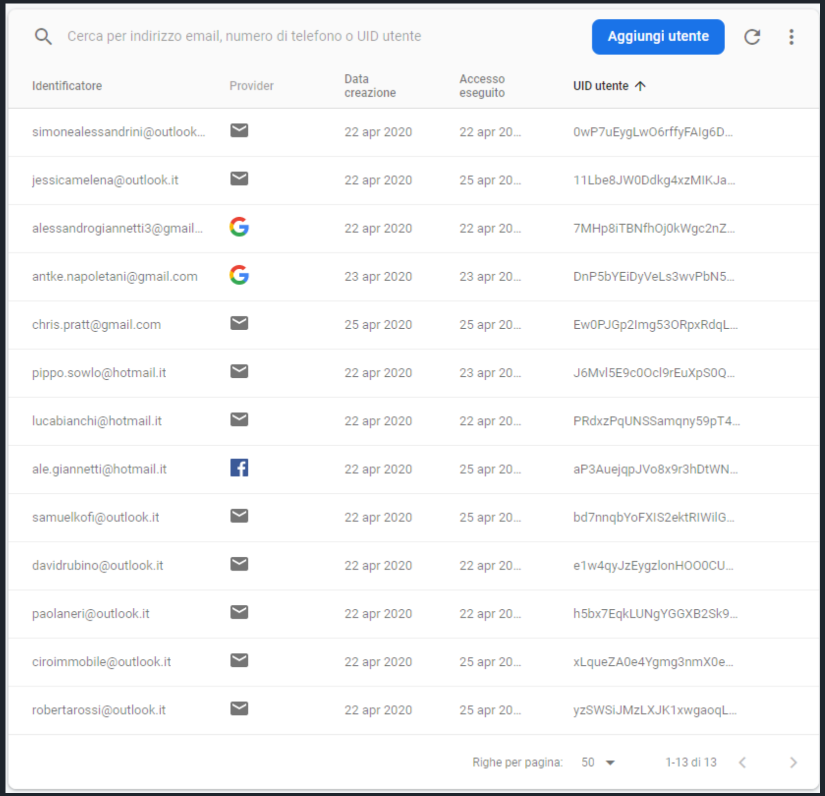


```
FoodLocker-API > app > models >
Project
    jobs
    mailers
    models
      concerns
      application_record.rb
      day.rb
      diary.rb
      food.rb
      food_day.rb
      post.rb
      relationship.rb
      sport.rb
      sport_day.rb
      user.rb
      weight.rb
```

Database


Weight
PK id: Integer
weight: float
photo: String
Like: integer
User
PK id: String
name: string
surname: string
photo: String
Post
PK id: Integer
title: string
content: string
photo: String
Like: integer
0..N
Register
1..1
1..1
Writes
0..N
0..N
0..N
Relationship
1..1
Owns
1..1
Diary
PK id: Intreger
1..1
Composed
0..N
Food_Day
PK id_food: integer
PK id_day: integer
meal: integer
0..N
Have
1..1
Day
PK id: Integer
date: date
water: integer
note: string
Have
1..1
0..N
Have
1..1
Sport_Day
PK id_sport: integer
PK id_day: integer
minute: integer
1..1
Have
0..N
0..N
Have
1..1
Food
PK id: integer
name: string
calories: Int
... all nutrition values
Sport
PK id: integer
name: string
calories: int

Firebase

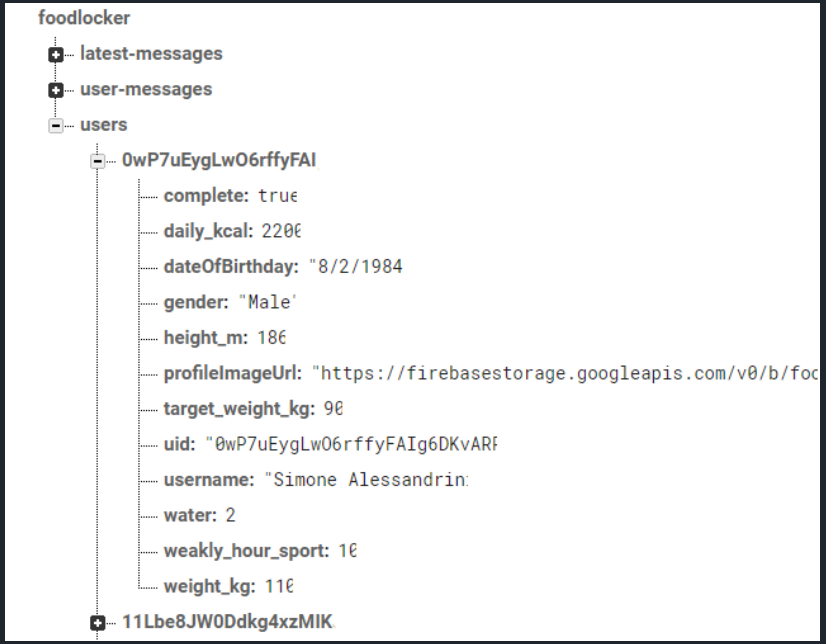


Cerca per indirizzo email, numero di telefono o UID utente

Aggiungi utente

Identificatore	Provider	Data creazione	Accesso eseguito	UID utente ↑
simonealessandrini@outlook...	✉	22 apr 2020	22 apr 20...	0wP7uEygLwO6rffyFAIg6D...
jessicamelena@outlook.it	✉	22 apr 2020	25 apr 20...	11Lbe8JW0Ddkg4xzMIKJa...
alessandrogiannetti3@gmail...	G	22 apr 2020	22 apr 20...	7MHp8iTBNfhOj0kWgc2nZ...
antke.napoletani@gmail.com	G	23 apr 2020	23 apr 20...	DnP5bYEiDyVeLs3wvPbN5...
chris.pratt@gmail.com	✉	25 apr 2020	25 apr 20...	Ew0PJGp2Img53ORpxRdqL...
pippo.sowlo@hotmail.it	✉	22 apr 2020	23 apr 20...	J6Mvl5E9c0Ocl9rEuXpS0Q...
lucabianchi@hotmail.it	✉	22 apr 2020	22 apr 20...	PRdxzPqUNSSamqny59pT4...
ale.giannetti@hotmail.it	f	22 apr 2020	25 apr 20...	aP3AuejqpJVo8x9r3hDtWN...
samuelkofi@outlook.it	✉	22 apr 2020	25 apr 20...	bd7nnqbYoFXIS2ektRIWilG...
davidrubino@outlook.it	✉	22 apr 2020	22 apr 20...	e1w4qyJzEygzlonHOO0CU...
paolaneri@outlook.it	✉	22 apr 2020	22 apr 20...	h5bx7EqkLUNgYGGXB2Sk9...
ciroimmobile@outlook.it	✉	22 apr 2020	25 apr 20...	xLqueZA0e4Ygmg3nmX0e...
robertarossi@outlook.it	✉	22 apr 2020	25 apr 20...	yzSWSiJMzLXJK1xwgaoqL...

Righe per pagina: 50 ▾ 1-13 di 13

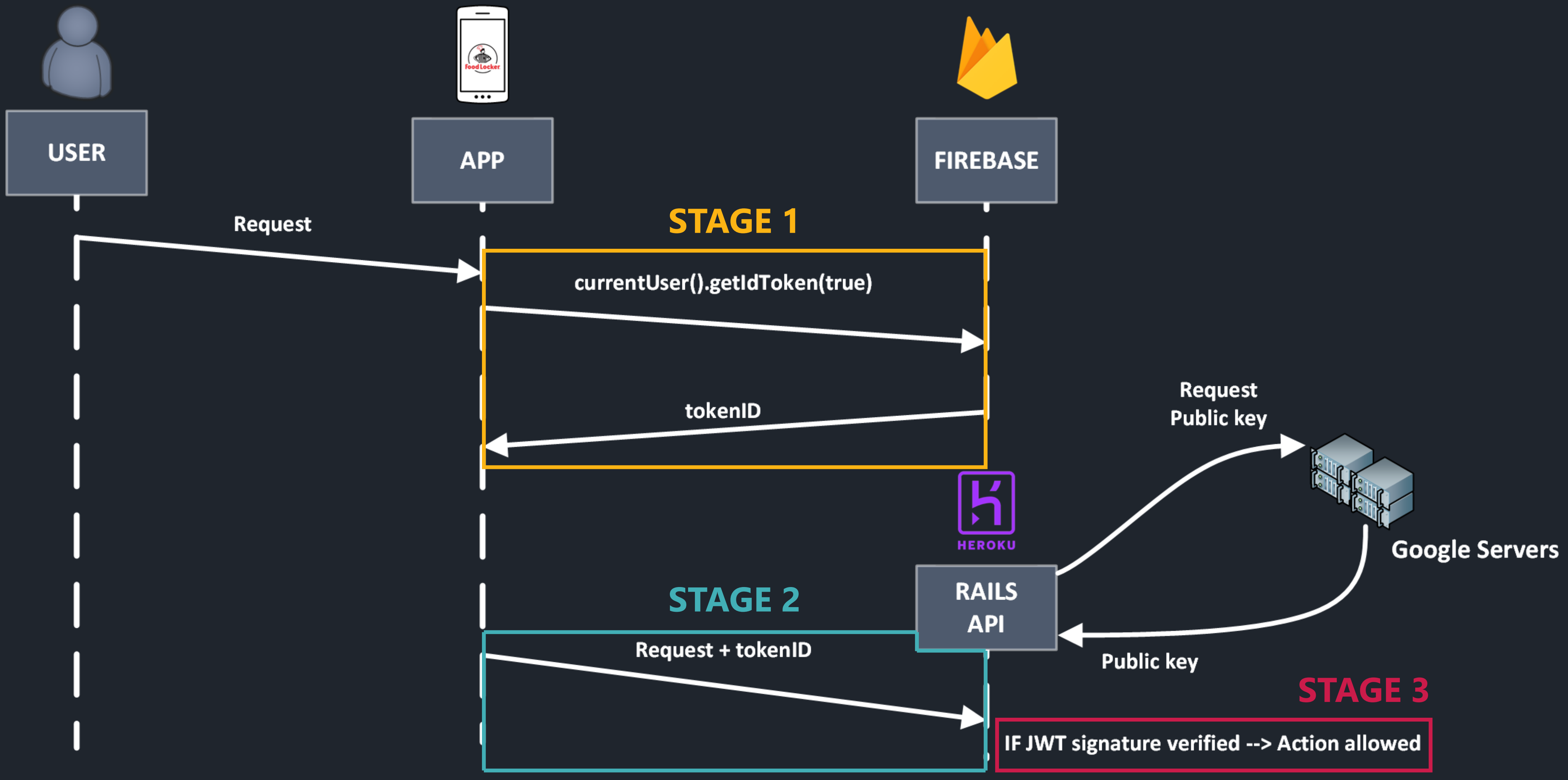


```
foodlocker
+ latest-messages
+ user-messages
- users
    - 0wP7uEygLwO6rffyFAI
        complete: true
        daily_kcal: 220
        dateOfBirthday: "8/2/1984
        gender: "Male'
        height_m: 186
        profileImageUrl: "https://firebasestorage.googleapis.com/v0/b/foc
        target_weight_kg: 90
        uid: "0wP7uEygLwO6rffyFAIg6DKvARI
        username: "Simone Alessandrin
        water: 2
        weakly_hour_sport: 10
        weight_kg: 110
    + 11Lbe8JW0Ddkg4xzMIK
```

Authentication page:
where data of registered users are stored

Realtime database:
Additional user information is stored here (weight, calories, height, wateretc.)
And messages that users exchange in chat

Token Authentication (Rails + Firebase)



Token Authentication (Rails + Firebase)

```kotlin
fun getTokenID(): String {
    var token: String? = ""
    auth.currentUser?.getIdToken(true)
            ?.addOnCompleteListener(OnCompleteListener { task ->
                if (!task.isSuccessful)
                    return@OnCompleteListener

                // Get new Instance ID token
                token = task.result?.token
                Log.e("token", token.toString())
            })
    return token.toString()
}
```

network/ApiService.kt

```kotlin
private val tokenID = FirebaseDBMng.getTokenID()

// assignment of the firebase token inside the header in the "Authorization" field
private var client = OkHttpClient.Builder().addInterceptor { chain ->
    val newRequest = chain.request().newBuilder()
            .addHeader("Authorization", tokenID)
            .build()
    chain.proceed(newRequest) }.build()

private val moshi = Moshi.Builder()
        .add(KotlinJsonAdapterFactory())
        .build()

private val retrofit = Retrofit.Builder()
        .addConverterFactory(MoshiConverterFactory.create(moshi))
        .addCallAdapterFactory(CoroutineCallAdapterFactory())
        .client(client)
        .baseUrl(BASE_URL)
        .build()
```

network/BackendRequestMng.kt

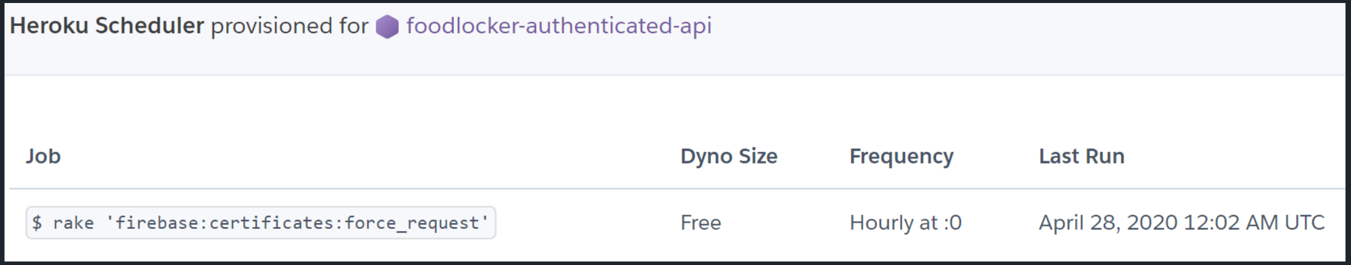
STAGE 1

Requesting the current user's tokenID to Firebase

STAGE 2

Construction of the Http request header via OkHttpClient, and assignment of the "Authorization" with the tokenID

Token Authentication (Rails + Firebase)



Heroku Scheduler provisioned for foodlocker-authenticated-api

Job	Dyno Size	Frequency	Last Run
`$ rake 'firebase:certificates:force_request'`	Free	Hourly at :0	April 28, 2020 12:02 AM UTC

```ruby
namespace :firebase do
 namespace :certificates do
  desc "Request Google's x509 certificates when Redis is empty"
  task request: :environment do
   FirebaseIdToken::Certificates.request
  end

  desc "Request Google's x509 certificates and override Redis"
  task force_request: :environment do
   FirebaseIdToken::Certificates.request!
  end
 end
end
```

lib/tasks/firebase.rake

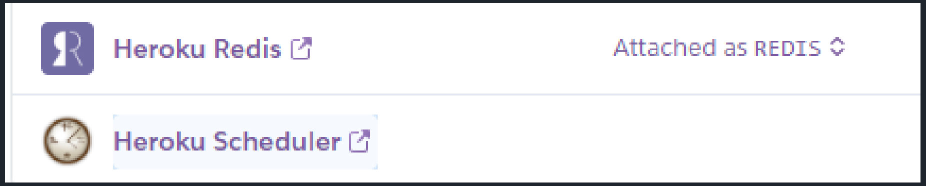
STAGE 3

To verify the tokenID of firebase we used a Gem called: **firebase_id_token**

3.1

In order to verify the token signature, we perform the request every hour of the google certificate through the heroku add-on "heroku-scheduler"

This certificate is stored in the Redis database, also implemented with the heroku add-on



Heroku Redis — Attached as REDIS

Heroku Scheduler

Token Authentication (Rails + Firebase)

```
class ApplicationController < ActionController::API
  before_action :authenticate_request
  attr_reader :current_user

  include Response
  include ExceptionHandler

  private

  def authenticate_request
    @current_user = AuthorizeApiRequest.call(request.headers).result
    render json: { error: 'Not Authorized' }, status: 401 unless @current_user
  end
end
```

controllers/application_controller.rb

STAGE 3

3.2

Before executing each request, the actual validity of the tokenID is checked in the "application_controller"

Token Authentication (Rails + Firebase)

```
[ ..... ]
 attr_reader :headers

 def user
   if decoded_auth_token.nil?
     errors.add(:token, 'Missing or Invalid token')
   else
     @user ||= User.find(decoded_auth_token[:user_id])
     if @user.nil?
       User.create!(id: @decoded_auth_token[:user_id],
               username: @decoded_auth_token[:name],
               photo_profile: @decoded_auth_token[:picture])
     end
   end
 end

 def decoded_auth_token
   @decoded_auth_token ||= FirebaseIdToken::Signature.verify(http_auth_header)
 end

 def http_auth_header
   if headers['Authorization'].present?
     return headers['Authorization'].split(' ').last
   else
     errors.add :token, 'Missing token'
   end
   nil
 end
[ ...... ]
```

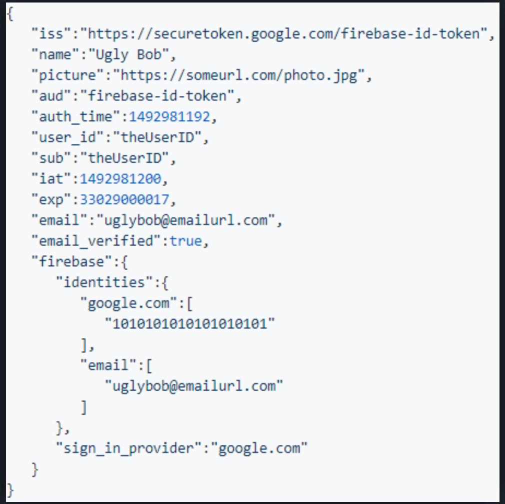
app/commands/authorize_api_request.rb

STAGE 3

3.3

Thanks to the gem you can check a token with the simple command:
`FirebaseIdToken::Signature.verify('token')`

which returns the token payload if it is verified, otherwise *nil*



```
{
   "iss":"https://securetoken.google.com/firebase-id-token",
   "name":"Ugly Bob",
   "picture":"https://someurl.com/photo.jpg",
   "aud":"firebase-id-token",
   "auth_time":1492981192,
   "user_id":"theUserID",
   "sub":"theUserID",
   "iat":1492981200,
   "exp":33029000017,
   "email":"uglybob@emailurl.com",
   "email_verified":true,
   "firebase":{
      "identities":{
         "google.com":[
            "1010101010101010101"
         ],
         "email":[
            "uglybob@emailurl.com"
         ]
      },
      "sign_in_provider":"google.com"
   }
}
```

Android Application

Material Design

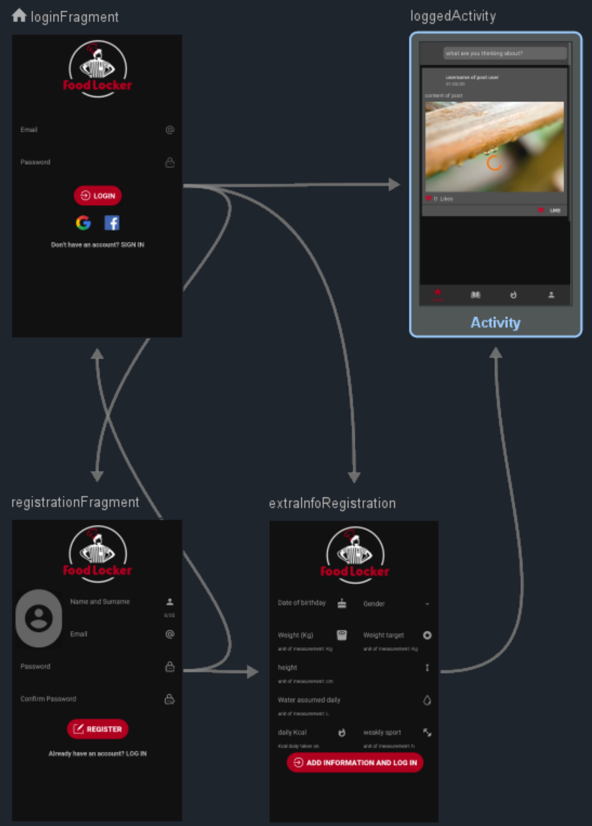
Responsive Layout

Multithread
Coroutine + retrofit for http request on separated thread
(MOSHI library used for conversion of json string to kotlin object)

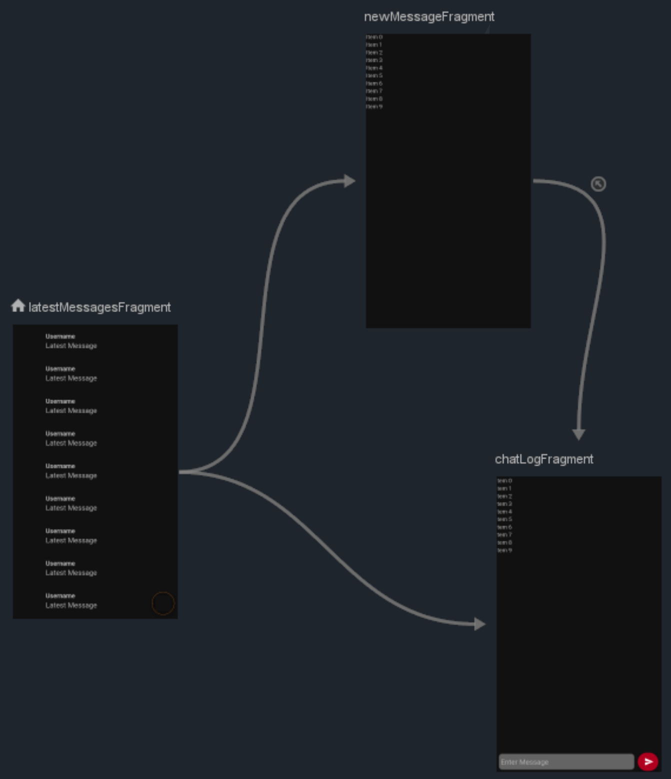
Model View ViewModel pattern design

Graphical representation of data
(chart implemented with MPAndroidChart)

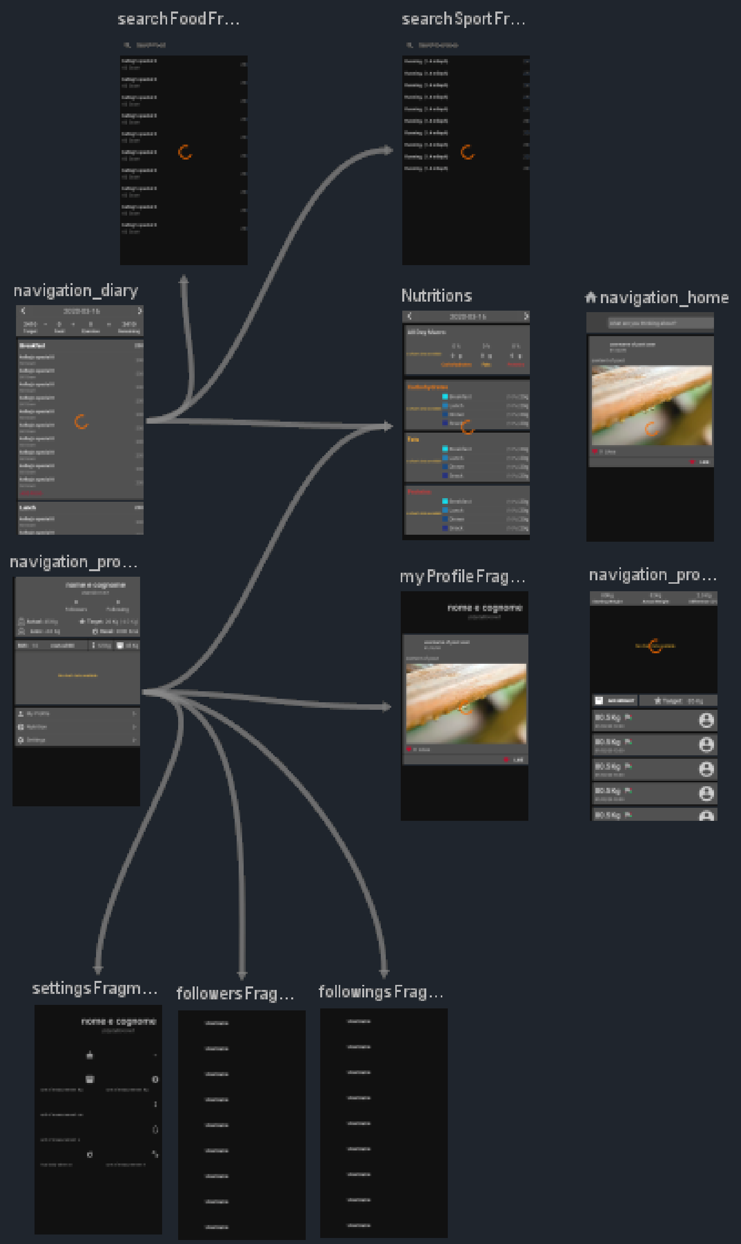
Navigation Path



Login Navigation Path



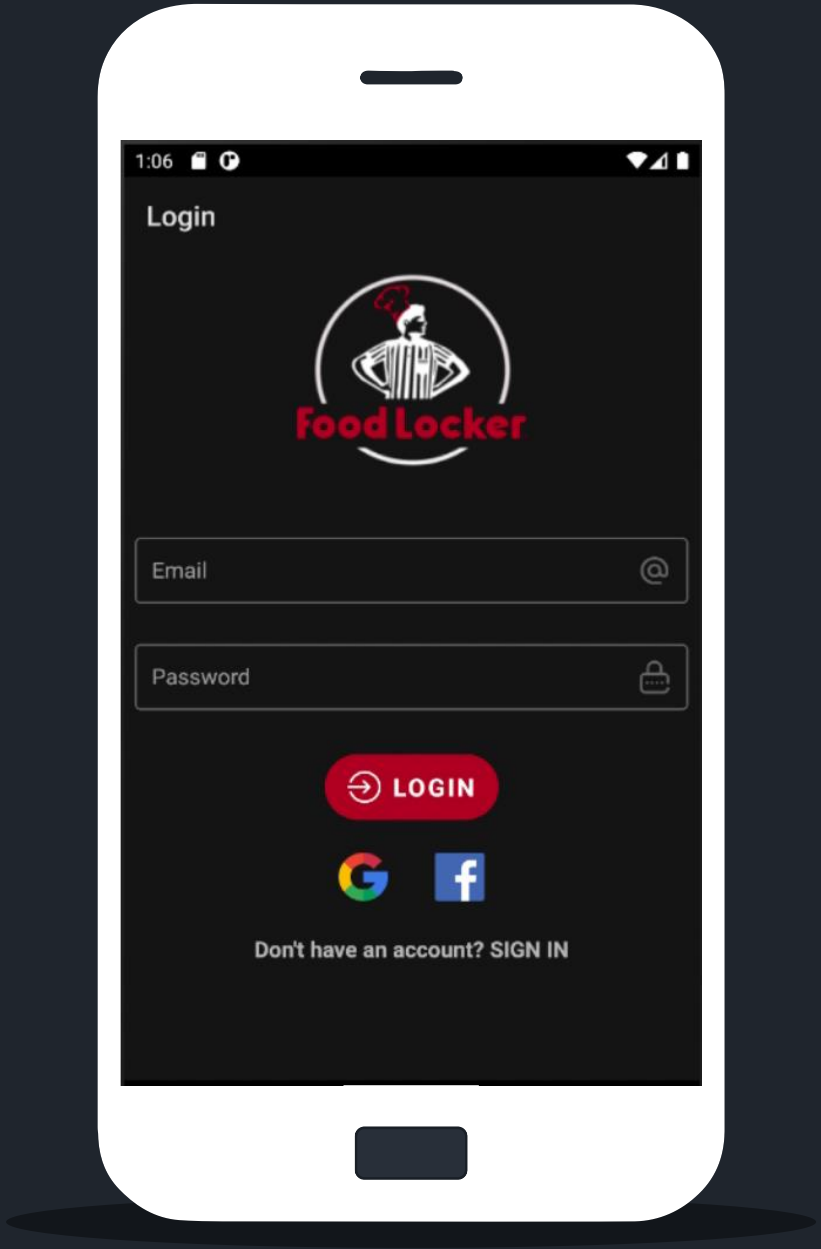
Messages Navigation Path



Main Navigation Path

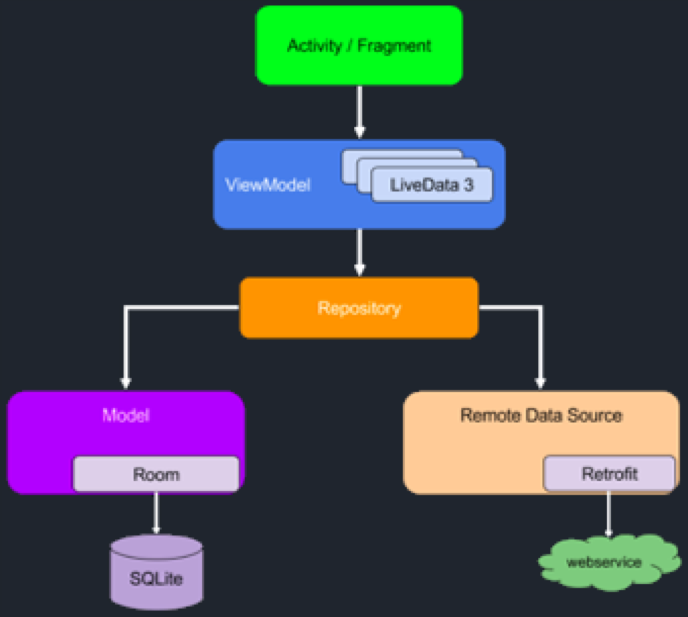
Login / Signup

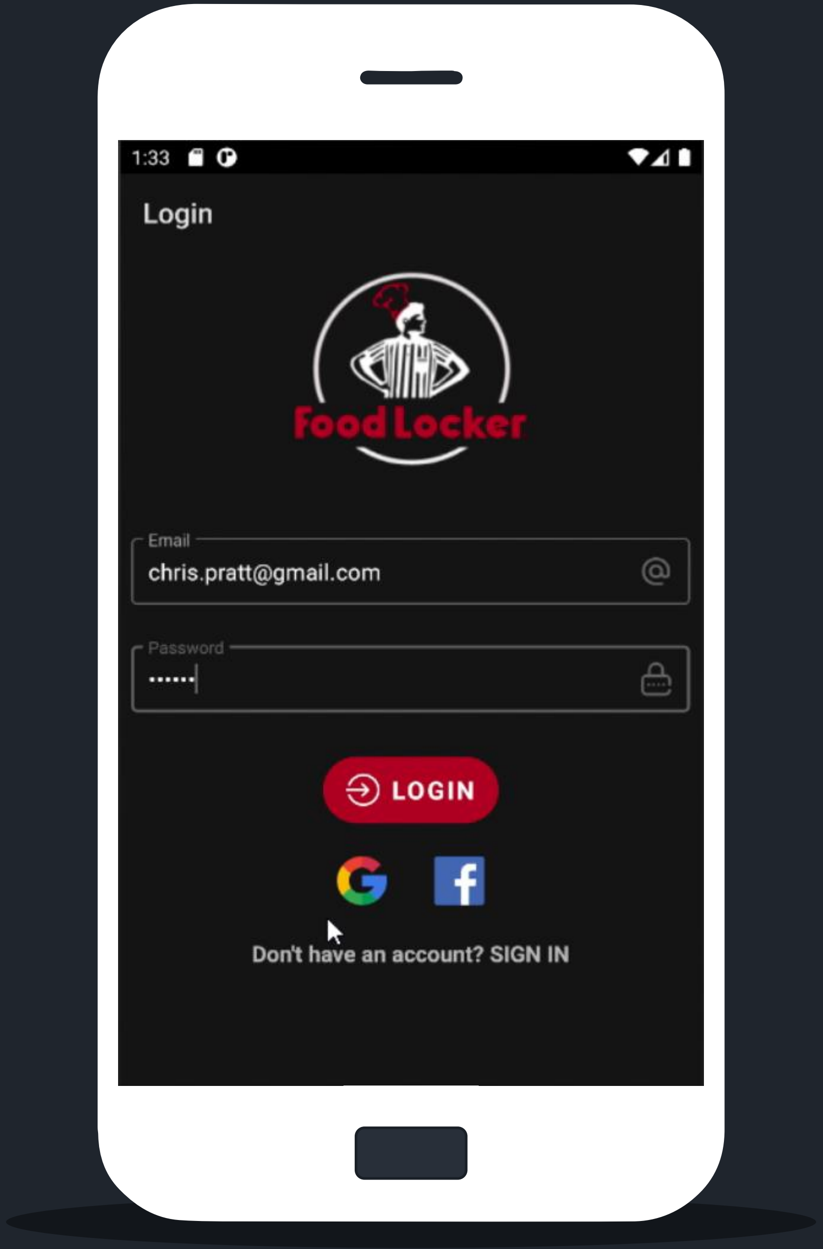
- Login method supported:
  - Email and Password
  - Facebook
  - Google
- If the user's profile is not complete, the user is redirected to the form to fill in the additional data



Home Page

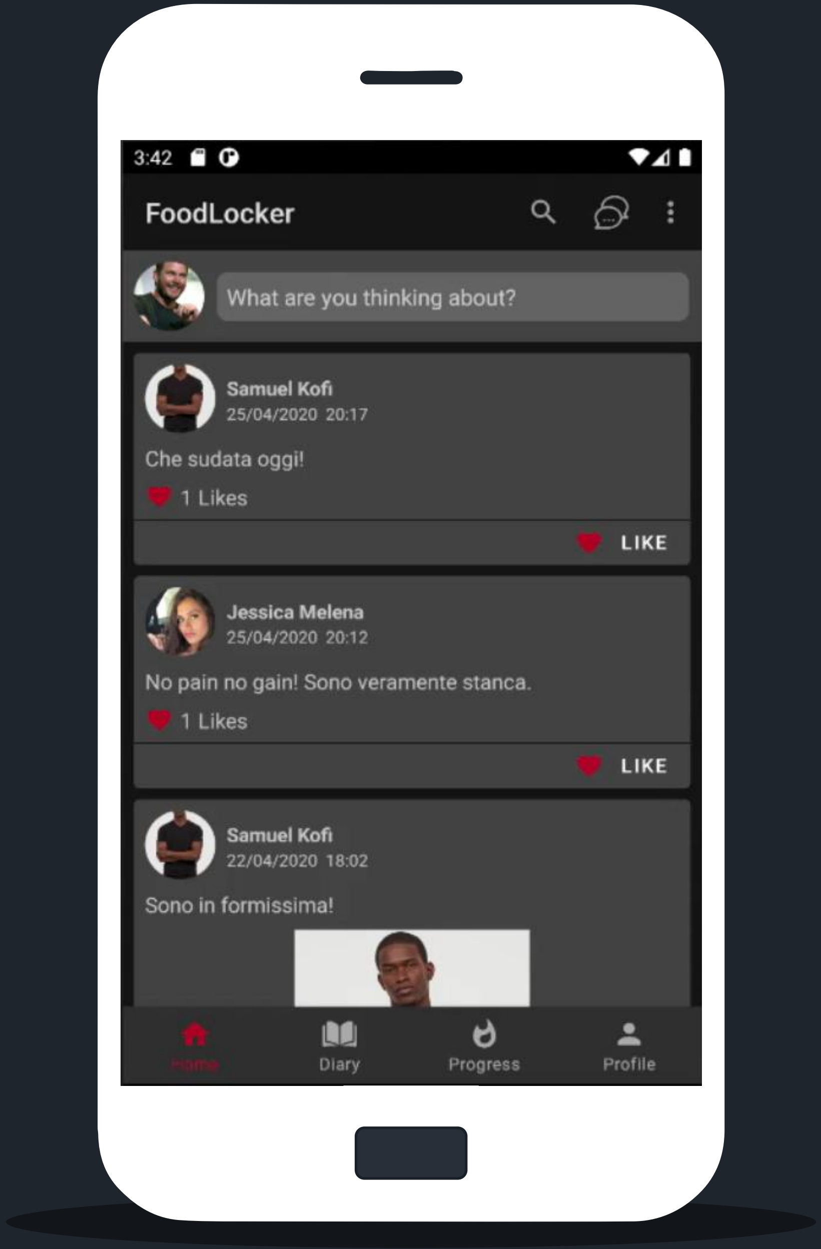
- See the list of posts of the people we follow and our personal posts, displayed by a Recycler View
- The posts are managed by a repository in order to manage a caching system
- For each post you can like it





Uploading a Post

- Add a new post
- By clicking the “publish" button the post is recorded via rails API on the Postgres database
- When an image is also attached, it is first uploaded to the Firebase storage, and once loaded the url is stored with the content of the post


3:42

FoodLocker

What are you thinking about?

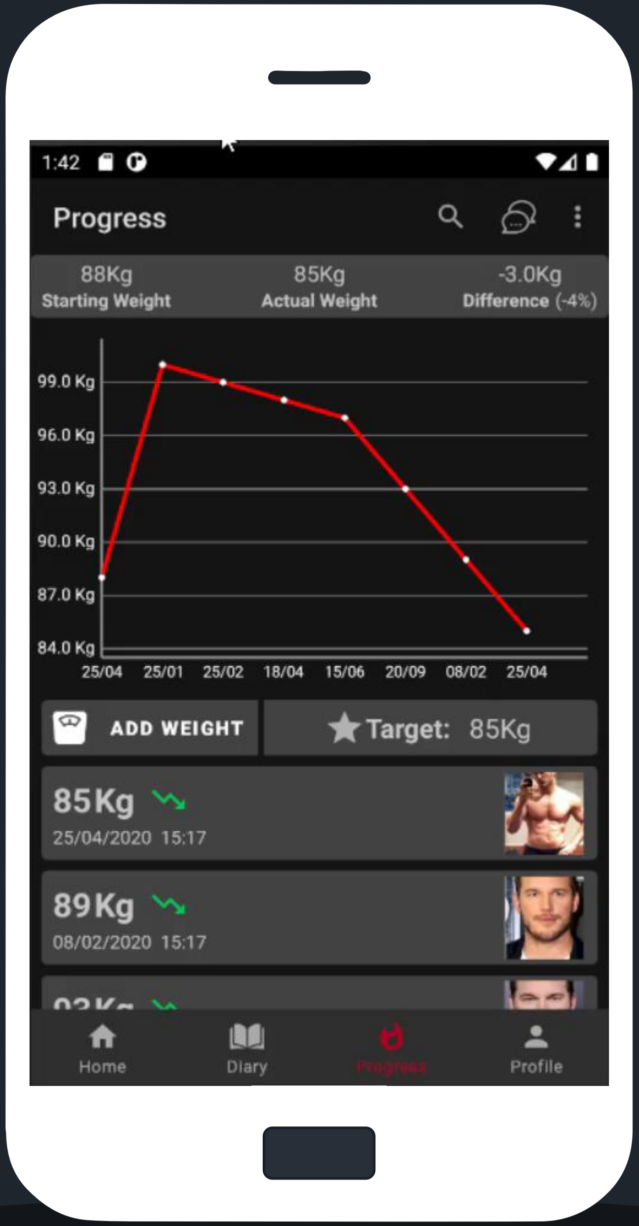
Samuel Kofi
25/04/2020 20:17
Che sudata oggi!
1 Likes
LIKE

Jessica Melena
25/04/2020 20:12
No pain no gain! Sono veramente stanca.
1 Likes
LIKE

Samuel Kofi
22/04/2020 18:02
Sono in formissima!

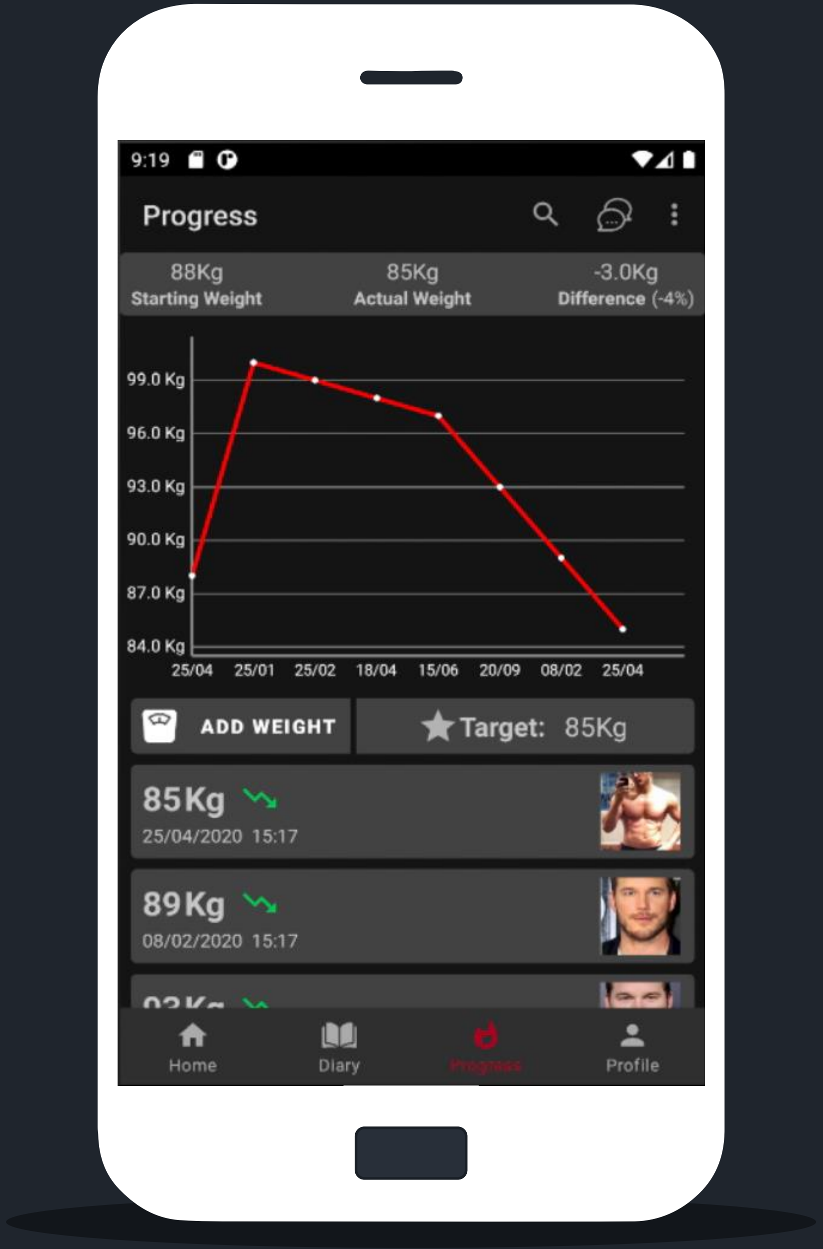
Progress

- Visualize the weighing trend thanks to a recycleview
- Plot the weights



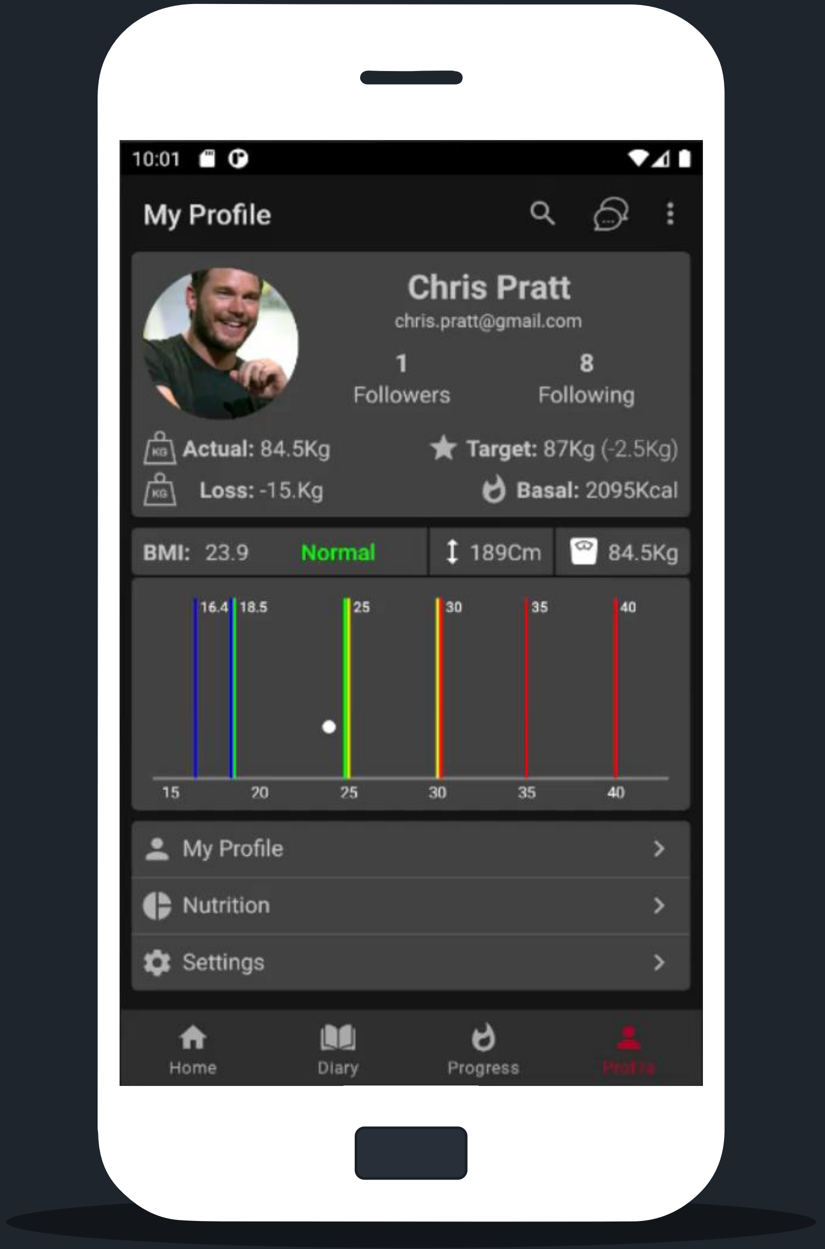
Uploading a Weight

- Add a new weight
- By clicking the "add weight state" button the weight is recorded via the rails API on the Postgres database
- When an image is also attached, it is first uploaded to the Firebase storage, and once loaded the url is stored with the weight



Profile

- Analyze your personal information, including the **BMI value**
- Check your followers or followed users through a recycleview
- Check your published posts through a recycleview
- Check your nutritional diary data
- Edit your personal data



10:01

My Profile

Chris Pratt
chris.pratt@gmail.com

1 Followers
8 Following

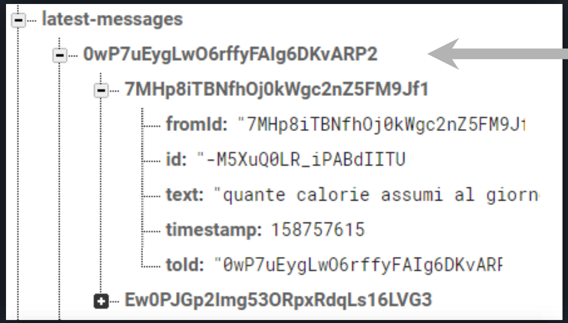
Actual: 84.5Kg
Target: 87Kg (-2.5Kg)
Loss: -15.Kg
Basal: 2095Kcal

BMI: 23.9 Normal
189Cm
84.5Kg

Chat

- The chat can be reached from any position via the button on the toolbar

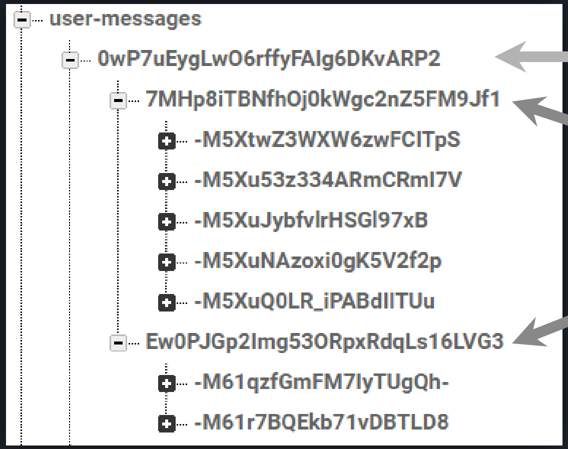
- For the implementation we used the firebase realtime database.



```
latest-messages
  0wP7uEygLw06rffyFAlg6DKvARP2
    7MHp8iTBNfhOj0kWgc2nZ5FM9Jf1
      fromId: "7MHp8iTBNfhOj0kWgc2nZ5FM9J1
      id: "-M5XuQ0LR_iPABdIITU
      text: "quante calorie assumi al giorn
      timestamp: 158757615
      toId: "0wP7uEygLw06rffyFAIg6DKvARF
    Ew0PJGp2Img53ORpxRdqLs16LVG3
```

Root:
User UID

Children:
Last messages of each user that wrote to the User UID. (Used For the population of the recycleview of the chat)

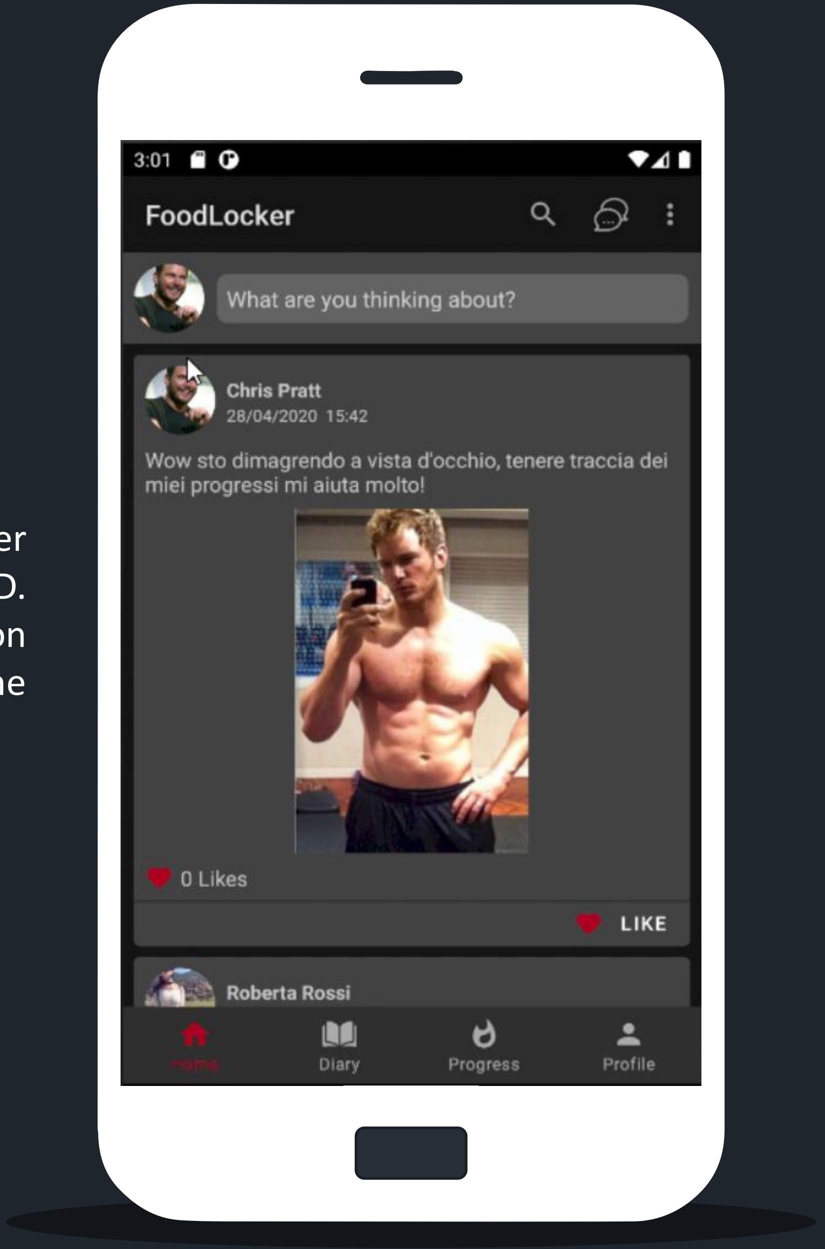


```
user-messages
  0wP7uEygLw06rffyFAlg6DKvARP2
    7MHp8iTBNfhOj0kWgc2nZ5FM9Jf1
      -M5XtwZ3WXW6zwFCITpS
      -M5Xu53z334ARmCRml7V
      -M5XuJybfvlrHSGl97xB
      -M5XuNAzoxi0gK5V2f2p
      -M5XuQ0LR_iPABdIITUu
    Ew0PJGp2Img53ORpxRdqLs16LVG3
      -M61qzfGmFM7lyTUgQh-
      -M61r7BQEkb71vDBTLD8
```

Root:
Chat log of User UID

Children:
Users you've chatted with

Sub Children:
Chat log



3:01

FoodLocker

What are you thinking about?

Chris Pratt
28/04/2020 15:42

Wow sto dimagrendo a vista d'occhio, tenere traccia dei miei progressi mi aiuta molto!

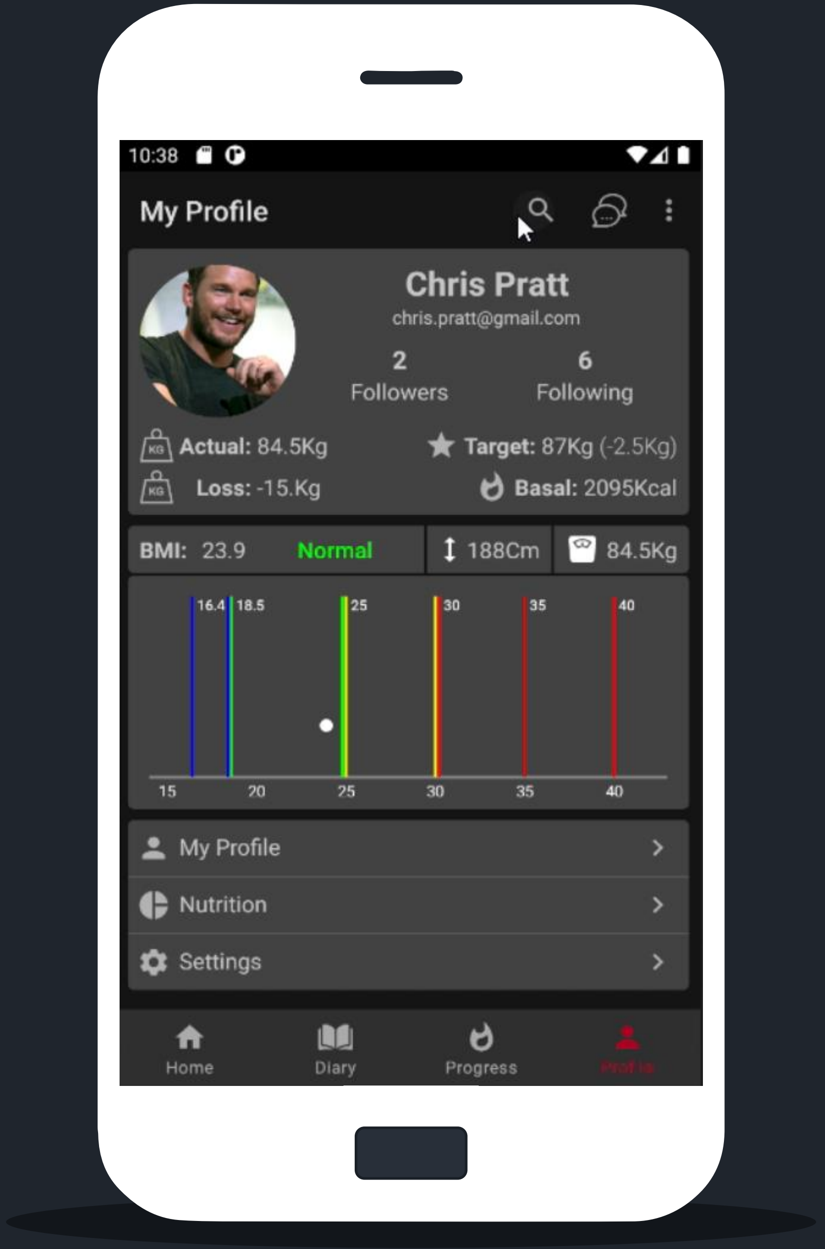
0 Likes

LIKE

Roberta Rossi

Search and follow Users

- Users can be searched from any position via the button on the toolbar
- The results of the search are shown thanks to a recycleview
- Follow a user , start a chat with him


10:38

My Profile

Chris Pratt
chris.pratt@gmail.com

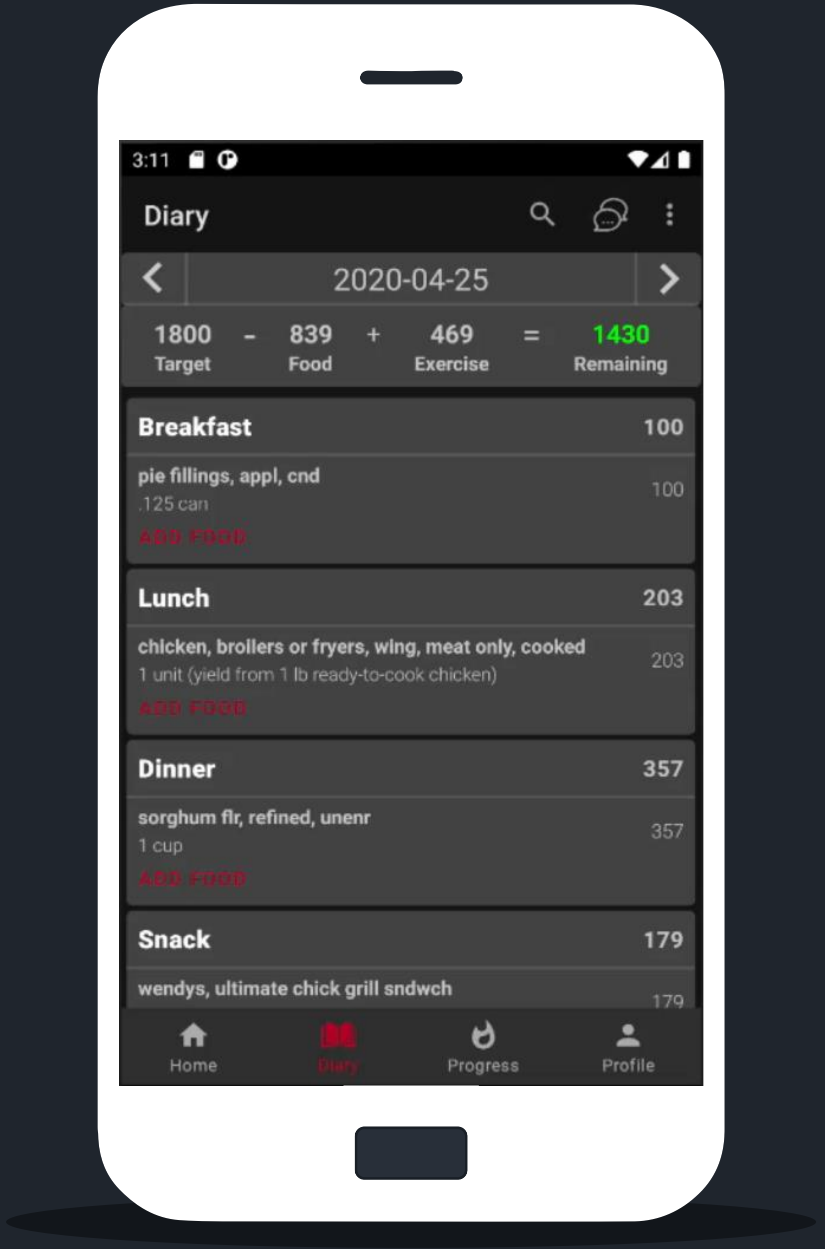
2 Followers
6 Following

Actual: 84.5Kg
Target: 87Kg (-2.5Kg)
Loss: -15.Kg
Basal: 2095Kcal

BMI: 23.9 Normal
188Cm
84.5Kg

Diary

- Navigate between the days of the diary
- Breakfast, Lunch, Dinner, Snack, Exercise details shown by different recycleviews
- See the count of the remaining calories
- Nutrition page show the total macronutrient intake count for the selected day and the macronutrient intake percentages per meal



2020-04-25

1800 Target - 839 Food + 469 Exercise = 1430 Remaining

Breakfast 100
pie fillings, appl, cnd 100
.125 can
ADD FOOD

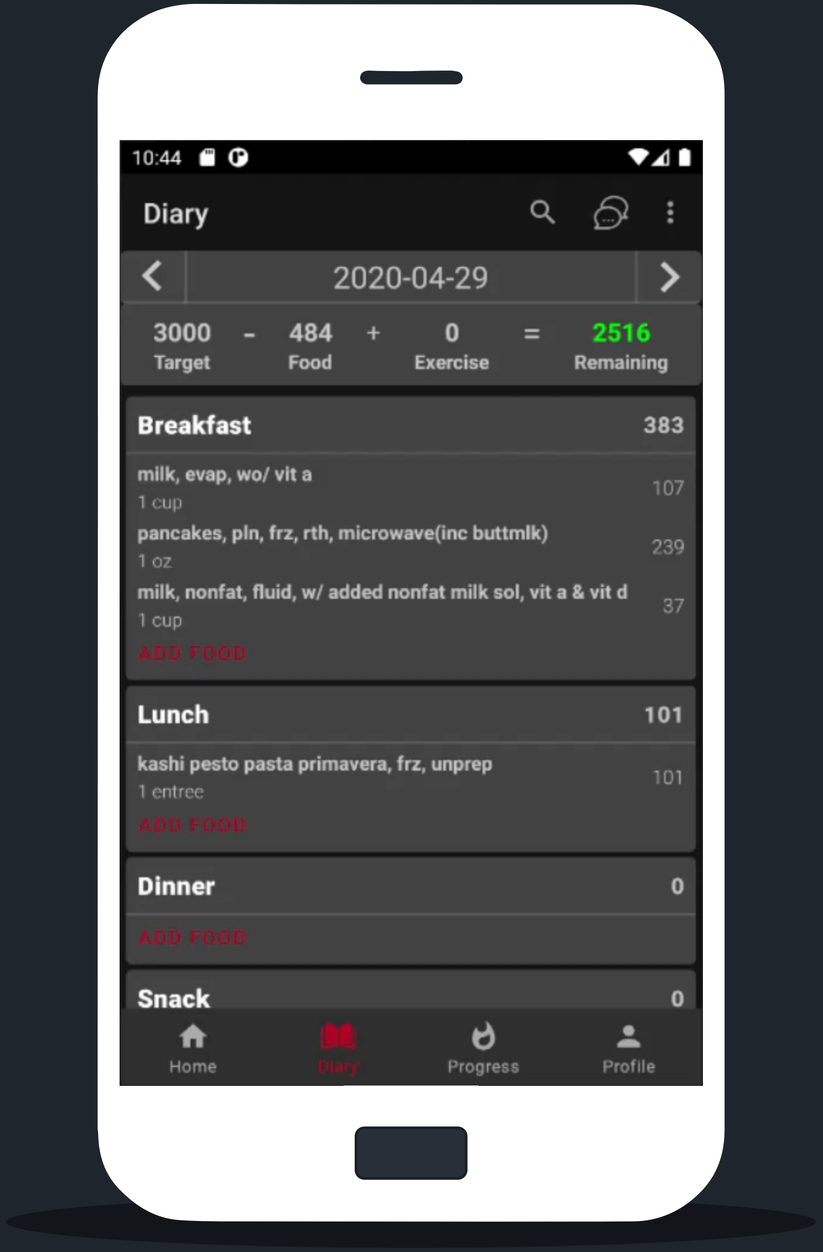
Lunch 203
chicken, broilers or fryers, wing, meat only, cooked 203
1 unit (yield from 1 lb ready-to-cook chicken)
ADD FOOD

Dinner 357
sorghum flr, refined, unenr 357
1 cup
ADD FOOD

Snack 179
wendys, ultimate chick grill sndwch 179

Uploading a Food entry

- Add a food for a specific meal
- Search food and display results through a recycleview
- Analyze macronutrients and nutritional table of the selected food


10:44

Diary

2020-04-29

3000	-	484	+	0	=	2516
Target		Food		Exercise		Remaining

Breakfast 383

milk, evap, wo/ vit a 107
1 cup

pancakes, pln, frz, rth, microwave(inc buttmlk) 239
1 oz

milk, nonfat, fluid, w/ added nonfat milk sol, vit a & vit d 37
1 cup

ADD FOOD

Lunch 101

kashi pesto pasta primavera, frz, unprep 101
1 entree

ADD FOOD

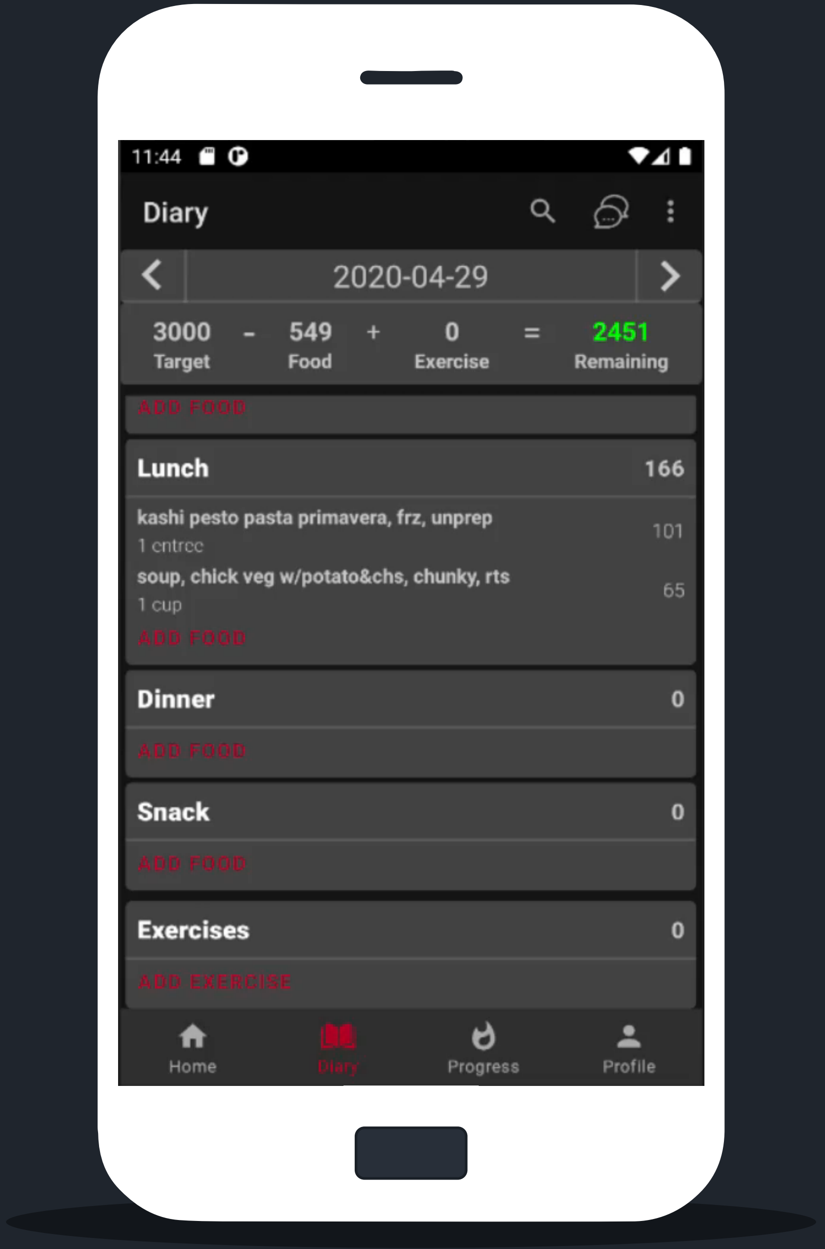
Dinner 0

ADD FOOD

Snack 0

Uploading a Sport entry

- Add an exercise
- Search an exercise and display results through a recycleview
- Calories are displayed thanks to live data



11:44

Diary

2020-04-29

3000	-	549	+	0	=	2451
Target		Food		Exercise		Remaining

ADD FOOD

Lunch 166

kashi pesto pasta primavera, frz, unprep 101
1 entree

soup, chick veg w/potato&chs, chunky, rts 65
1 cup

ADD FOOD

Dinner 0

ADD FOOD

Snack 0

ADD FOOD

Exercises 0

ADD EXERCISE

THANKS FOR THE ATTENTION