

Project Εξαμήνου (40%)

Να δημιουργήσετε μια εφαρμογή flutter, στην οποία θα κάνετε οπτικοποίηση συγκεκριμένων Fitbit δεδομένων, με χρήση του πακέτου charts_flutter (https://pub.dev/packages/charts_flutter). Περισσότερες λεπτομέρειες για τα γραφήματα θα βρείτε στο link <https://google.github.io/charts/flutter/gallery.html> . Συγκεκριμένα, μέσω του Fitbit Web API Swagger (<https://dev.fitbit.com/build/reference/web-api/explore/>), να κάνετε Authorize τα απαραίτητα Scopes και στη συνέχεια να κάνετε POST requests και να εισάγετε δεδομένα, μέσω του Swagger, για Sleep Data (POST) για 7 ημέρες :

POST `/1.2/user/-/sleep.json` Log Sleep

Έπειτα, να εκτελέσετε GET requests μέσω του Swagger, έτσι ώστε να πάρετε αυτά τα δεδομένα και να τα κάνετε download σε μορφή .json. Επίσης, να κάνετε download τα δεδομένα που αφορούν το προφίλ του χρήστη, με το αντίστοιχο Swagger GET request.

1. Sleep Data List

GET `/1.2/user/-/sleep/list.json` Get Sleep Logs List

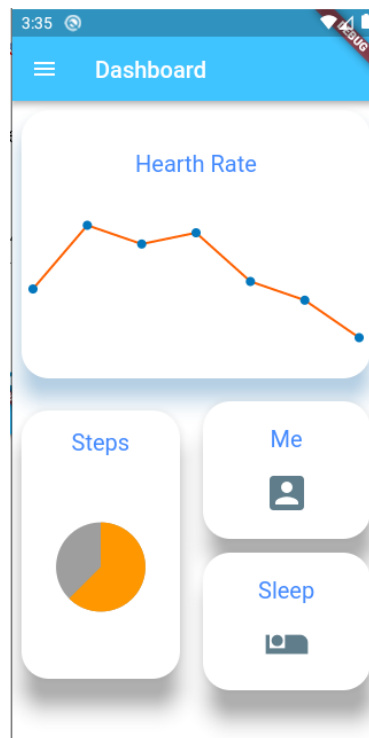
2. User Data

GET `/1/user/-/profile.json` Get Profile

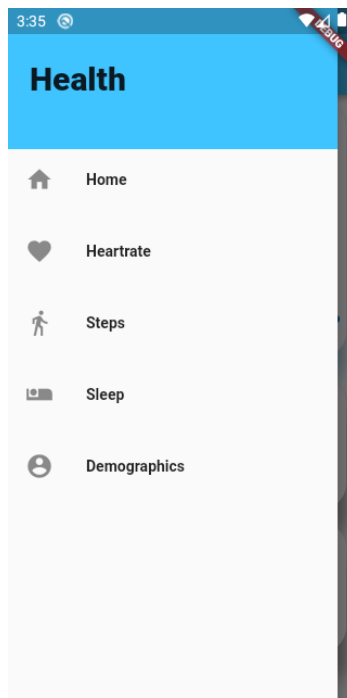
Παράδειγμα:



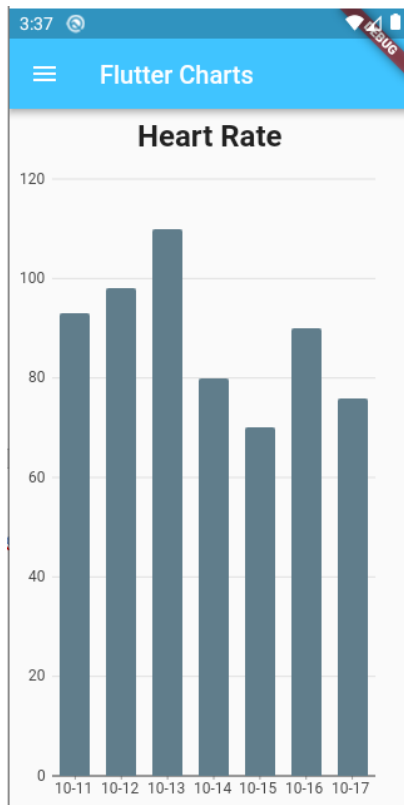
Η αρχική σελίδα της εφαρμογής θα είναι της μορφής:



Θα περιέχει ένα custom drawer της μορφής:



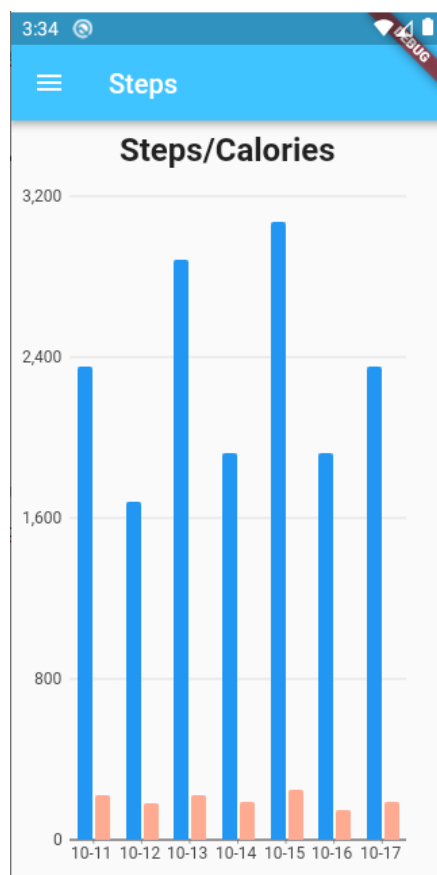
Στη σελίδα Home το γράφημα για το Heart Rate θα περιέχει τα πραγματικά δεδομένα από το αρχείο heartrate.json τα οποία δεδομένα θα είναι κοινά και για την οθόνη Heart Rate που θα είναι της μορφής:



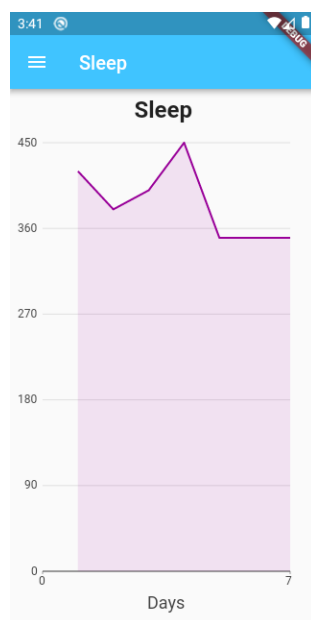
Στο γράφημα του steps στην αρχική οθόνη θα χρησιμοποιήσετε την τιμή για τα steps της τελευταίας ημερομηνίας για το πορτοκαλί κομμάτι του γραφήματος και το υπόλοιπο γκρι κομμάτι θα είναι η διαφορά των steps από την τιμή του ημερήσιου στόχου βημάτων. Θεωρούμε τον ημερήσιο στόχο για τα βήματα τα 8000 βήματα και θέλουμε να δείξουμε στο γράφημα την διαφορά από τα 8000 βήματα.

Στο e-class θα βρείτε ένα αρχείο "response_calories_steps.json", το οποίο περιγράφει την δραστηριότητα του χρήστη (User Activity Data) για επτά συγκεκριμένες ημερομηνίες και περιέχει, μεταξύ άλλων, τιμές για Calories και Steps. Θα χρησιμοποιήσετε αυτές τις τιμές για την οθόνη Steps

Παράδειγμα:

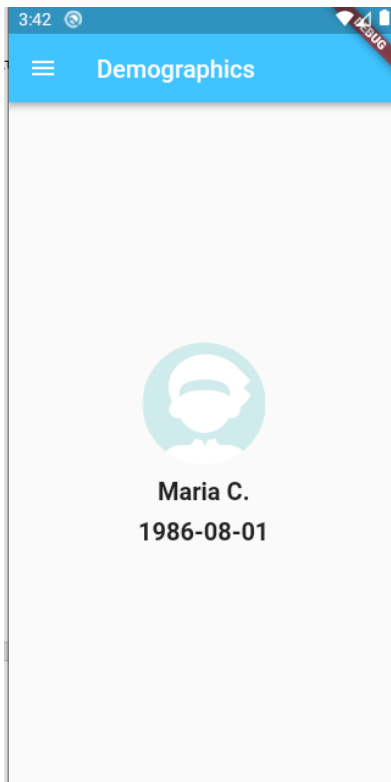


Στην οθόνη sleep θα δημιουργήσετε ένα γράφημα που να δείχνει τις τιμές για τις 7 ημέρες που περάσατε δεδομένα στο swagger. Παράδειγμα :



Τέλος στην καρτέλα me/demographics θα εισάγετε τα στοιχεία του χρήστη. Το avatar που χρησιμοποιείτε, όνομα και ημερομηνία γέννησης.

Παράδειγμα:



Σημαντική λεπτομέρεια το sleep είναι σε milliseconds θα πρέπει να το μετατρέψετε σε λεπτά.

1. Να παρουσιάζονται οι βασικές πληροφορίες από το προφίλ του χρήστη (Όνομα, Ηλικία, Avatar).
2. Να γίνει οπτικοποίηση για τα Steps και Calories με χρήση ενός Grouped Bar Chart (https://google.github.io/charts/flutter/example/bar_charts/grouped)
3. Να γίνει οπτικοποίηση των δεδομένων heart rate του χρήστη για μία εβδομάδα, με γράφημα της επιλογής σας από την βιβλιοθήκη **charts_flutter** χρησιμοποιώντας το heartrate.json
4. Να γίνει οπτικοποίηση των δεδομένων ύπνου του χρήστη (User Sleep Data) για 7 ημέρες, με γράφημα της επιλογής σας από την βιβλιοθήκη **charts_flutter**.
5. Το styling στις εικόνες είναι ενδεικτικό μπορείτε να χρησιμοποιήσετε όποιο styling θέλετε
6. Στην αρχική σελίδα (**ΚΑΙ ΜΟΝΟ ΣΤΗΝ ΑΡΧΙΚΗ**) μπορείτε να χρησιμοποιήσετε τα γραφήματα που διδαχθήκαμε στο εργαστήριο

2020-2021

7. Τα δεδομένα που θα χρησιμοποιήσετε στην εφαρμογή θα πρέπει να τα διαβάζετε από τα αρχεία json (Εκτός αν χρησιμοποιήσετε κλήση API).

➤ +20% εάν η εφαρμογή flutter επικοινωνεί με το Web API της Fitbit και παίρνει τα δεδομένα από εκεί, αντί από τοπικά .json αρχεία (εκτός την περίπτωση για το heart rate που μπορείτε να χρησιμοποιήσετε το αρχείο heartrate.json) .

➤ +10% για το πιο προσεγμένο και καλαίσθητο dashboard.

➤ Τα projects να ανέβουν στο e-class σε .zip μορφή

➤ Τέλος, να περιέχονται όλα τα αρχεία .dart , οι φάκελοι με assets (εάν υπάρχουν) της εφαρμογής όπως και το pubspec.yaml αρχείο. Για όσους κάνουν κλήσεις στο API από την εφαρμογή, θα πρέπει να εμπεριέχονται και τα απαραίτητα αρχεία με τα δεδομένα που χρειάζονται για την διασύνδεση της εφαρμογής με το API.

Σημείωση: Ο κώδικας να είναι σωστά στοιχισμένος, να περιέχει σχόλια και μία σύντομη περιγραφή πάνω από κάθε function.