

Food Habit Tracker

Matti Bragge

Kuvaus ohjelmasta

Food Habit Tracker on hiilijalanjäljen pienentämisessä auttava hyvinvointiapplikaatio, johon käyttäjä voi syöttää omia tietojaan saadakseen arvion tuottamastaan kasvihuonekaasumäärästä ruokakulutuksen osalta. Lisäksi ohjelma sisältää mahdollisuuden kirjata ja seurata oman painon kehitystä esimerkiksi painonpudotuksen apuna. Ohjelmaan voi kirjautua usealla eri käyttäjällä, jotta useampi henkilö voi seurata tietojensa kehitystä samalla laitteella.

Ohjelman ajatuksena on auttaa käyttäjää parantamaan ruokailutottumuksiaan antamalla tietoa keskivertosuomalaisen kulutuksesta ruokatyypeittäin sekä näyttämällä visuaalisesti graafina oman painon ja päästöjen kehityksen.

Ohjelmassa on viisi eri näkymää: Kirjautumis-, CO₂-päästöjen tarkastelu-, kirjaus-, painon tarkastelu-, ja käyttäjätietojen asetusnäkymä. Näkymät on eroteltu siten, että eri toiminnot ovat selkeästi erillään ja että käyttöliittymä ei tunnu ahtaalta.

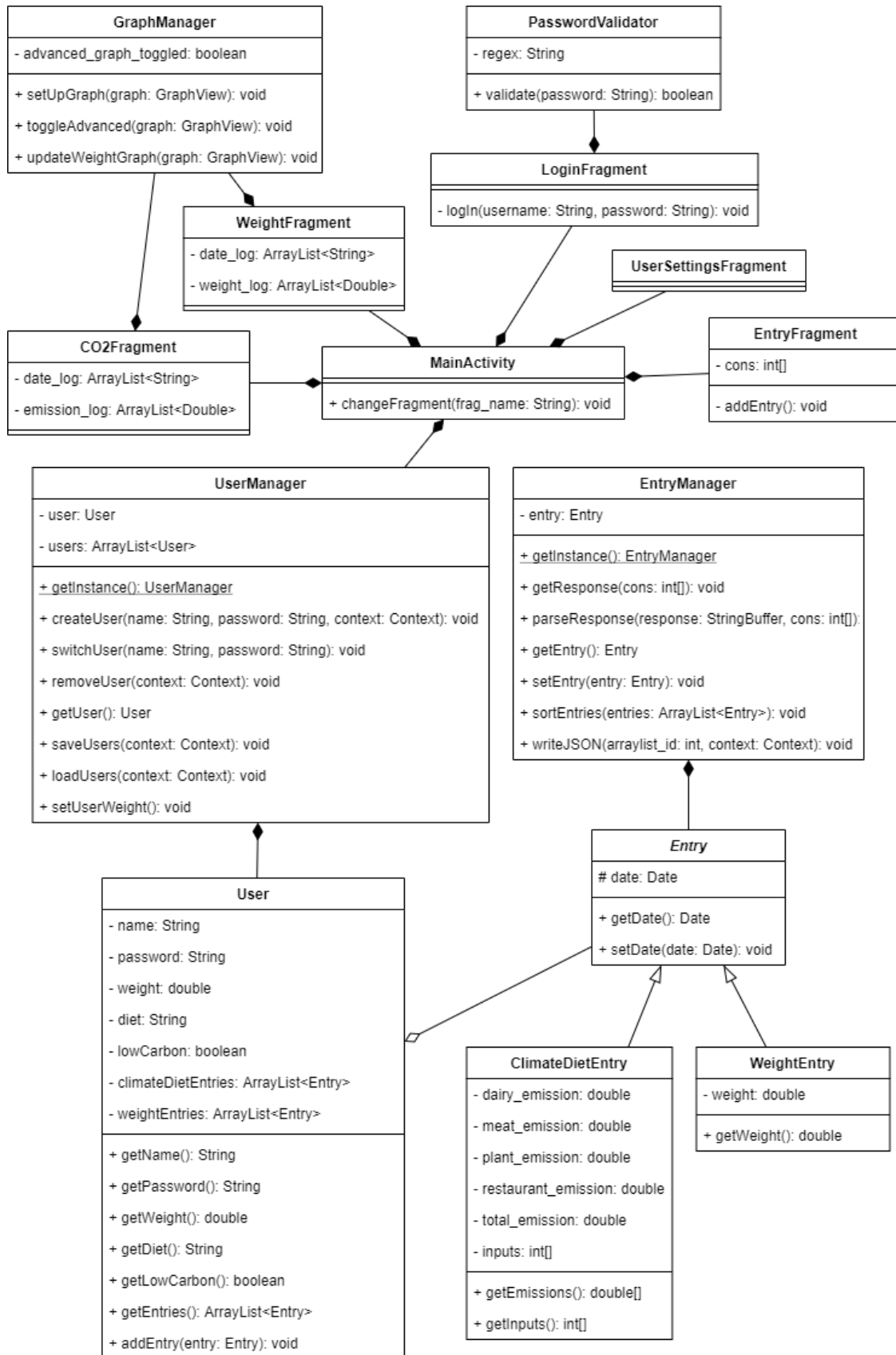
Tekijät

Tein harjoitustyön yksilötyönä.

Ohjelman toteutus

Ohjelma toimii Androidilla (API level 23 eli 6.0 tai uudempi). Se käyttää graafien piirtämiseen ulkoista kirjastoa GraphView (<https://github.com/jjoe64/GraphView>). Arvioitujen päästömäärien laskemiseen ohjelma käyttää ilmastodieetin julkista rajapintaa, FoodCalculator:ia. Käytetyt työkalut ovat draw.io luokkakaavion piirtämiseen, Figma käyttöliittymäsuunnitelman tekemiseen, Android Studio itse ohjelman kehitykseen ja testaukseen, GitHub versionhallintaan sekä Microsoft Excel ajankäytön kirjaamiseen.

Luokkakaavio



Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Olio-ohjelmoitu	Ohjelma on ohjelmoitu Javalla ja noudattaa olio-ohjelmoinnin periaatteita	13 / 5
Vähintään viisi erilaista luokkaa & oliota	Ohjelma sisältää 8 luokkaa käyttöliittymäluokkia lukuun ottamatta sekä >8 oliota	13 / 5
Vähintään yhden API:n käyttö	Ohjelma käyttää ilmastodieetin FoodCalculator-rajapintaa	13 / 5
Käyttäjän toiminnan tallentaminen logiin	Ohjelma tallentaa käyttäjäoliot (jotka sisältävät logit ja muut tiedot) tiedostoon. Ohjelmassa on myös mahdollisuus tallentaa käyttäjän logit erillisenä JSON-tiedostona	13 / 5
Login tarkastelumahdollisuus graafina tai tekstinä	Logia voi tarkastella graafina ja tekstinä ohjelman sisällä	13 / 5
Hyvin suunnitellut UI-komponentit	UI-komponentit on aseteltu ja tyylitelty sen mukaisesti, että käyttökokemus on mahdollisimman kitkaton ja miellyttävä	1-5
Kirjautuminen applikaatioon	Ohjelmassa on mahdollista kirjautua sisään käyttäjätiliin	3
Sovelluksella voi olla useampi käyttäjä, käyttäjien luominen, tietojen tallennus	Ohjelmassa on mahdollista luoda useampi käyttäjätili, jotta useampi henkilö voi käyttää ohjelmaa samalla laitteella. Tiedot tallentuvat ja latautuvat automaattisesti	3
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä	Ohjelma ei anna käyttäjän luoda tiliä, ellei salasana täytä kriteerejä (väh. 12 merkkiä, >1 pieni kirjain, >1 iso kirjain, >1 erikoismerkki (@#\$%))	2
Perustietojen syöttö (paino, ruokavalio, vähäpäästöisten ruokien suosiminen) ja näiden arvojen käyttö	Ohjelmassa on oma näkymä erikseen tallennettaville käyttäjätiedoille. Näitä tietoja käytetään painon kehityksen kirjaamisessa ja päästöjen laskennassa	2
Ohjelma kerää käyttäjän massan kehityksestä dataa ja näyttää muutokset graafisesti havainnollistaen ruudulla	Ohjelmaan voi syöttää painon ja nähdä sen kehitys ajan kuluessa graafilla sekä listassa	3
Ohjelma näyttää graafisesti ilmastodieetin tarjoamien	Ohjelmassa voi painaa nappulaa näyttääkseen graafissa totaalipäästöjen lisäksi myös eri päästöarvojen kehityksen	3

arvojen muutokset käppyröillä		
Fragmenttien hyödyntäminen aktiviteettien sijaan käyttöliittymässä	Ohjelma käyttää fragmentteja eri näkymiin vaihtamisessa	2
Scoped storagen käyttäminen tiedon tallennuksessa	Ohjelma ei vaadi käyttäjän myöntämiä oikeuksia laitteen massamuistiin	2
Responsiivinen käyttöliittymä	UI-komponentit käyttävät ConstraintLayout:ia, jotta ruudun koon muuttuessa ne skaalautuvat ja liikkuvat järkevästi	2
Summa		36-40

Työmäärät

Tekijä	Tehtävät	Tunnit
Matti Bragge	Devaus, testaus, suunnittelu, dokumentointi	35
Summa		35

Mitä opin harjoitustyöstä?

Opin käyttämään graafeja, käyttämään julkista rajapintaa ja luomaan kirjautumisjärjestelmän olio-ohjelmoinnin mukaisesti. Interfacen käyttäminen projektissani paransi käsitystäni niiden ideasta. Lisäksi sain paremman kuvan kokonaisen applikaation luomisesta prosessina.

Palaute harjoitustyöstä

Harjoitustyön tekeminen oli mielekästä, sopivan haastavaa ja aihe oli kiinnostava. Mikään implementoimani ominaisuus ei ollut liian hankala muihin verrattuina, mutta eniten aikaa meni suunnittelun ja toteutuksen osalta hiilikuormakirjauksen tekemisominaisuuteen. Joihinkin vaihtoehtoihin ominaisuuksiin (esim. salasanan hashaukseen) olisi voinut olla jonkinlaista apumateriaalia linkattuna.