

<Food Calculator>

Tekijät - Tomi Rahkonen - 0506950

Kuvaus ohjelmasta

- Mitä ohjelma sisältää

Ohjelma laskee arvion käyttäjän vuoden aikana kerryttämille hiilidioksidipäästöille perustuen. Syötteeksi annetaan viikon aikana kulutetut määrät eri ravinnonlähteille.

Käyttäjä syöttää ohjelmaan viikon aikana syömänsä ateriat raaka-aineiden mukaan (grammoina), minkä jälkeen ohjelma vertaa niitä suomalaiseen keskiarvoon ja syöttää arvot (prosentteina) Ilmastodieetti-API:lle. Tämän jälkeen käyttäjä saa nähtäville arvioidun ruoasta aiheutuvien CO₂-päästöjensä määrän vuoden aikana.

Lisäksi ohjelmassa on mahdollisuus selata vanhoja tuloksia, jolloin voidaan nähdä esimerkiksi päästöjen kokonaismäärän muutos sen jälkeen, kun kulutustottumuksia on muutettu. Tämän lisäksi ohjelmassa on mahdollisuus luoda ja käyttää useita eri käyttäjiä, joille tulee myös lisätä (vahva) salasana. Tietojen tallentamiseen käytettiin SQLite-tietokantaa.

- Miksi nämä asiat on sisällytetty ohjelmaan?

Vanhojen tuloksien seuraamiseksi on järkevää luoda käyttäjälle käyttäjätunnus. Kun käyttäjätunnuksen luonti on lisätty sovellukseen, samalla vaivalla saadaan luotua mahdollisuus usealle käyttäjälle tietokannan avulla. Yksityisyyden vuoksi on myös järkevää lisätä salasana ja vahvan salasan vaatiminen lisää turvallisuutta, eikä sen sisällyttäminen juuri lisää sovelluksen monimutkaisuutta sillä tietokanta on jo toteutettu käyttäjiä varten.

SQLiten hyödyntäminen sisällytettiin ohjelmaan, koska SQL:n käytöstä on jo kiitettävästi aiempaa kokemusta joten se oli minulle yksinkertaisin tapa sisällyttää kyseiset ominaisuudet sovellukseen.

Tekijät

- Kuka teki ja mitä?
- Kuvaus työnjaosta ja rooleista

“Ryhmääni” ei kuulunut muita jäseniä, eli tein työn yksin.

Ohjelman toteutus

- Millaisella teknisellä alustalla ohjelma toimii?

Työ toimii Android 11.0 alustalla. Työn aikana ohjelman toimivuuden testaamiseen käytetty laite oli Pixel 2 5,0”, 1080x1920.

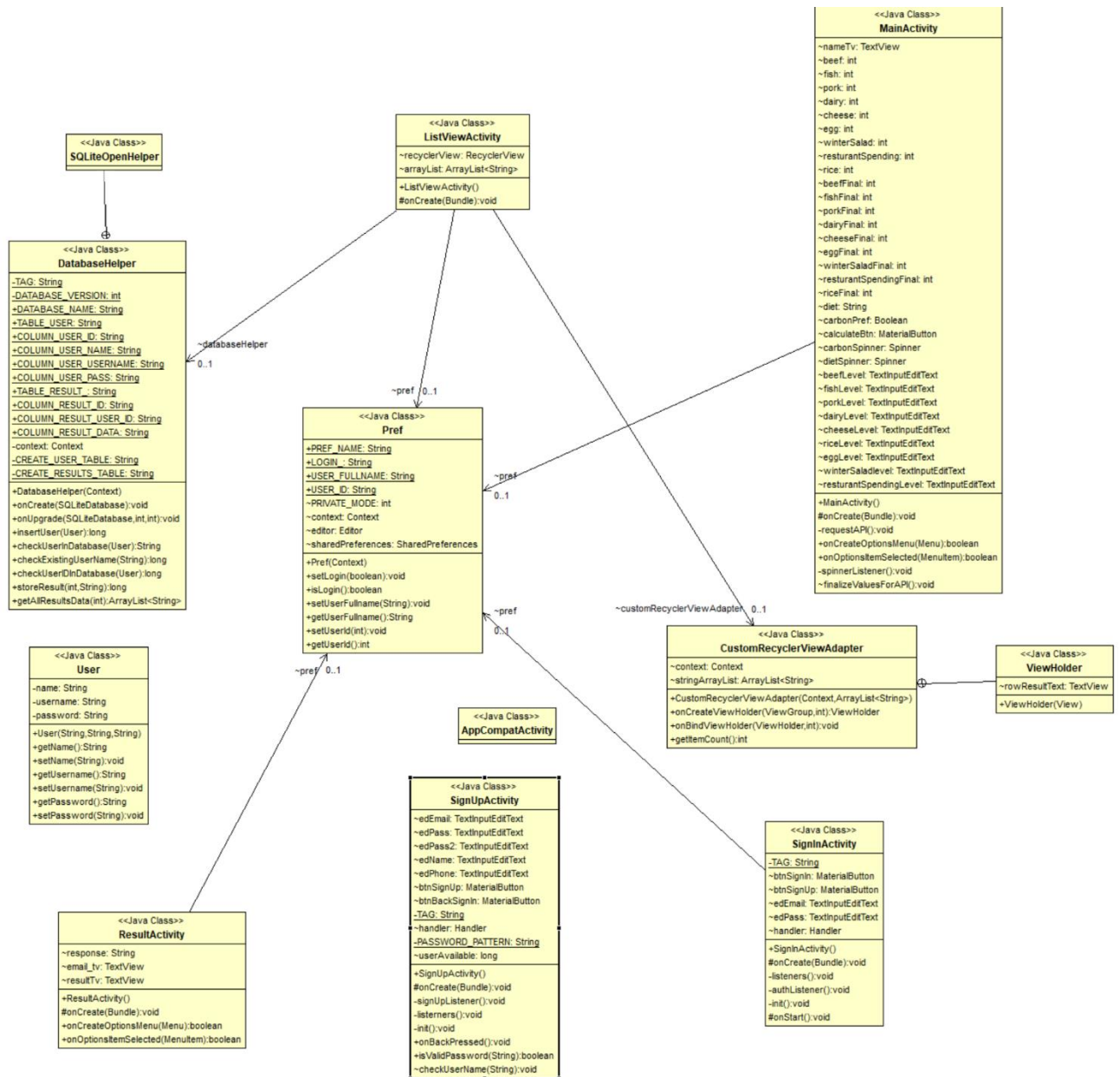
- Mitä kirjastoja on käytetty?

com.android.volley
androidx.appcompat
androidx.constraintlayout:constraintlayout
com.google.android.material

- Mitä työkaluja on käytetty?
 - Ryhmätyökalut
 -
 - Ohjelmistokehitystyökalut
 - Android Studio Bumblebee/2021.1.1
 - Eclipse IDE 4.18
 - Testaustyökalut
 -
 - Dokumentointityökalut
 - Diagrams.net
 - GitHub

Luokkakaavio

Löytyy myös erikseen .png-tiedostona repositorysta osana dokumentaatiota.



Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Olio-Ohjelmoitu	(Pakollinen)- Toteutuu	-
Vähintään viisi erilaista luokkaa & oliota (käyttöliittymäluokkia ei lasketa)	(Pakollinen) - Sovelluksessa 9 erilaista luokkaa & oliota	-
Sovellus tallentaa käyttäjän	(Pakollinen) –	-

toiminnan (käyttäjän syöttämät arvot / tulokset) logiin (JSON, XML jne.)	Toteutettu käyttäen SQLitea. Itselle helpoin ratkaisu johtuen aikaisemmasta kokemuksesta SQL:n käytössä.	
Logia on mahdollista tarkastella (puhtaana tekstinä, graafisilla käppyröillä jne.), eli voidaan tutkia arvojen (esim. oma massa) kehitystä kirjausten edetessä	Pakollinen – Mahdollisuus avata "historia", jolloin listattuna kaikki aikaisemmat tulokset. Ei toteutettu käppyröitä tai vastaavaa hifistelyä, mutta toteutus mielestäni vähintäänkin täyttää vaatimuksen sillä on kattavampi kuin vain puhdas teksti. Helppo seurata esimerkiksi tulosten kehitystä.	-
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	"Hyvin suunniteltu" on aika subjektiivista. Eiköhän tuo yhden pisteen arvoinen ole.	1 (1-5)
Kirjautuminen	Ohjelma sisältää rekisteröitymisen ja kirjautumisen	3 pistettä
Sovelluksella voi olla useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin	Sovelluksella voi olla (käytännössä) lukematon määrä käyttäjiä ja sovellus sisältää rekisteröitymisen. Tietojen tallennus SQL-tietokannan avulla.	3 pistettä
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä)	Salasanan vaatimukseksi asetettu 12-20 merkkiä, minkä lisäksi vähintään yksi numero, erikoismerkki sekä iso/pienet kirjaimet.	2 pistettä
Ohjelma muistaa käynnistämisen / kirjautumisen jälkeen missä näkymässä käyttäjä oli ennen ohjelman sulkemista	Toteutettu vaatimuksen edellyttämällä tavalla	2 pistettä
Responsiivinen käyttöliittymä (toimii siis erikokoisilla ruuduilla sulavasti)	Toteutettu vaatimuksen edellyttämällä tavalla	2 pistettä
Summa		13+13 = 26

Työmäärät

Tekijä	Tehtävät	Tunnit
Tomi Rahkonen	Sovelluksen koodaaminen	n. 80?
Tomi Rahkonen	Dokumentointi, luokkakaavio ym.	3
	Työ aloitettu jo kesäkurssin aikana 2021, jatkettu loppuun kevään 2022 kurssilla. Kokonaistyömäärää hankala arvioida tästä johtuen, mutta varmaan jotain tuota luokkaa.	
Summa		80-85

Mitä opin harjoitustyöstä?

Merkittävimpänä asiana työssä opin käyttämään Android Studiota, joka oli minulle vielä täysin tuntematon. Javasta on jonkin verran kokemusta ja SQL vielä tutumpi. Näiden käyttö yhdessä kuitenkin uutta.

developer.android.com sekä sen sisältämien oppaiden hyödyntämienn. Näistä sai äärimmäisen hyvän selityksen ja esimerkkikoodin moneen työssä käyttämäni toteutukseen, jotka ensin vaikuttivat monimutkaisilta.

Palaute harjoitustyöstä (vapaaehtoinen)

- Mitkä ominaisuudet / toiminnot olivat helppoja / vaikeita toteuttaa?

Mielestäni yleisesti tuo kirjautuminen ja käyttäjien luonti (mukaanlukien salasanan määrittelyt ym.) tarjoavat aika paljon pisteitä suhteessa niiden vaikeuteen. Toisaalta moni listan loppupään toiminnoista vaikutti turhan vaikealta pisteisiin nähden, joten ne jäivät toteuttamatta. Tämä voi tosin olla ihan tarkoituksenmukaistakin.

- Oliko jokin asia aivan syvältä?
 - Android Studion ja Java-versioiden kanssa kikkailu, etenkin kun jatkoin keskeneräistä työtä pitkän tauon jälkeen. Lopputuloksena oli koko Android Studion ja Javan poistaminen koneelta ja uudelleenasetus.
 - Luokkakaavio
- Oliko jokin asia todella hyvää tässä työssä?

Riittävän haastava. Harjoitustyö vaati oikeasti hieman perehtymistä, toisin kuin monet muut kurssit.

- Mitä toivoisit ensi vuoden harjoitustyöhön?