

OLIO HT

Tekijät:

Santeri Röpelin, Veikka Immonen ja Atte Lihtamo

Kuvaus ohjelmasta

- Ohjelma on tehty käyttäjän hiilijalanjäljen seuraamista varten, lisäksi käyttäjä pystyy seuraamaan painoaan. Ohjelma käynnistyy avatessa kirjautumisnäkympään, jossa pystyy kirjautumaan sisään tai luomaan uuden käyttäjän. Sisäänkirjautumisen jälkeen avautuu kotinäkymä, missä käyttäjä näkee hieman статистиikkaa omasta hiilijalanjäljestään. Näkymästä on myös mahdollista nappia painamalla lisätä uusia matkoja. Sivuvalikosta löytyy kotinäkymän lisäksi kuvaajat, asetukset ja uloskirjautuminen. Kuvaajista löytyy lisää статистиikkaa käyttäjän syötteistä. Kuvaajat hiilijalanjäljen ja painon kehityksestä, sekä kuljetuista kilometreistä. Kuljetut kilometrit näkyvät myös tekstimuodossa. Asetuksista pystyy muokkaamaan käyttäjäprofiilia ja tarkastelemaan logia tekstimuodossa. Ohjelma on myös helposti laajennettavissa uusilla ominaisuuksilla/asetuksilla.
- Asiat on sisällytetty ohjelmaan, jotta siitä tulee toimiva kokonaisuus, jota on helppo käyttää. Lisäksi tehtävänannossa määritettiin tietyt vaatimukset, ja tavoitteleimme hyvää arvosanaa, joka vaikutti ohjelman sisältöön.

Tekijät

- Santeri teki kirjautumiseen ja dataan liittyvän koodin back-end puolelle. Veikka koodasi front-end puolen LoginActivityyn ja oli apuna Santerille back-end puolella. Atte teki front-endin MainActivityyn puolelle.
- Suunnitteluun osallistuivat kaikki, heittelimme ideoita, joiden pohjalta lähdimme rakentamaan kokonaisuutta. Tehtävänanto määritteli pitkälle sen, mitä ohjelmaan sisällytämme. Roolit muodoistuivat itsestään mielenkiinnon ja osaamisen mukaan. Koska teimme työtä yhdessä discordissa keskustellen, työmäärät ovat kaikilla pitkälti samat ja samalla autoimme toisiamme ongelma kohdissa.

Luokkakaavio

Omana tiedostonaan.

Ohjelman toteutus

- Ohjelma toimii Android 6.0 tai uudemmalla järjestelmällä
- Käytetyt ulkoiset kirjastot:
 - Room <https://developer.android.com/training/data-storage/room>
 - Google Charts <https://developers.google.com/chart>
- Mitä työkaluja on käytetty?
 - Github, Discord
 - Android Studio, VS Code
 - Microsoft Word

Toteutetut ominaisuudet

Ominaisuus	Miksi	Pisteet
Olio-ohjelmoitu	Pakollinen	13
Vähintään viisi erilaista luokkaa & oliota (käyttöliittymäluokkia ei lasketa)	Pakollinen	
Vähintään yhden API:n käyttö, esim. Ilmastodieetti: https://ilmastodieetti.ymparisto.fi/ilmastodieetti/swagger/ui/index	Pakollinen	
Sovellus tallentaa käyttäjän toiminnan (käyttäjän syöttämät arvot / tulokset) logiin (JSON, XML jne.)	Pakollinen	
Logia on mahdollista tarkastella (puhtaana tekstinä, graafisilla käppyröillä jne.), eli voidaan tutkia arvojen (esim. oma massa) kehitystä kirjausten edetessä	Pakollinen	
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	Täytyy näyttää hyvältä	5 pistettä
Kirjautuminen applikaatioon	Turvallisuuden takia ja että sovellusta voi käyttää pitkäjäksoisesti (tiedot tallentuu)	3 pistettä
Sovelluksella voi olla useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin	Useampi ihminen pystyy käyttämään sovellusta	3 pistettä

Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä)	Turvallisuuden takia	2 pistettä
Salasanan tallennus käyttää jonkinlaista hash-menetelmää ja suolausta (esim SHA-512 + salt)	Turvallisuuden takia	2 pistettä
Ohjelmaan on mahdollista syöttää perustiedot (esim. pituus, paino, ikä(/syntymävuosi), kuva, asuinkunta) käyttäjästä ja näitä arvoja käytetään jossakin	Käyttäjä pystyy tallentamaan tietonsa ja seuraamaan niitä	2 pistettä
Ohjelma kerää käyttäjän massan kehityksestä dataa ja näyttää muutokset graafisesti havainnollistaen ruudulla	Visuaalisesti on helpompi seurata muutoksia kuin tekstistä.	3 pistettä
Ohjelma näyttää graafisesti ilmastodieetin tarjoamien arvojen muutokset käppyröillä (esim. kuinka lihan kulutus ja hiilijalanjälki on muuttunut aikojen saatossa)	Visuaalisesti on helpompi seurata muutoksia kuin tekstistä.	3 pistettä
Asynkronisten HTTP-kutsujen käyttö dataa haettaessa	Jotta ohjelma ei hidastu.	2 pistettä
Fragmenttien hyödyntäminen aktiviteettien sijasta käyttöliittymiä rakennettaessa	Yksinkertaisempi	2 pistettä
Scoped storagen käyttäminen tiedon tallennuksessa (ei vaadi käyttäjän myöntämiä oikeuksia laitteen massamuistiin, vaan toimii omassa "hiekkalaatikossaan")	Jotta ei vaadi käyttöoikeuksia	2 pistettä
Responsiivinen käyttöliittymä (toimii siis erikokoisilla ruuduilla sulavasti)	Eri käyttäjät käyttävät erikokoisia laitteita	2 pistettä
OMAT OMINAISUUDET		
Ohjelma viestii käyttäjälleen lyhyitä viestejä perustuen käyttäjän toimintaan	Hauska ominaisuus	1 piste
Suunnittelu	Helpompi lähteä koodaamaan	1 piste

Palautus ajoissa		1 piste
Yhteensä		47 pistettä

Työmäärät

Tekijä	Tehtävät	Tunnit
Santeri Röpelin	Back-end, testaus	n. 60
Veikka Immonen	Front-end + Back-end päivittelyä, testaus	n. 60
Atte Lihtamo	Front-end, testaus, dokumentaatio	n. 60
Summa		n. 180

Mitä opin harjoitustyöstä?

Santeri: Opin, miten salasanoja hashataan ja miten tietoa voi salata eri algoritmien avulla. Opin uusia ominaisuuksia Javasta, esim. käyttämään lambdaa ja runnablea. Uutena asiana tuli myös androidin Room:in hyödyntäminen. Lisäksi opin versionhallinnan perusteet ja ymmärrän nyt Android-sovellusten rakenteen hyvin.

Veikka: Opin käyttämään githubia, kuinka androidsovelluksia tuotetaan ja miten API:a voidaan hyödyntää. Opin laajasti front-end puolta, kuten xml-tiedostojen tekoa ja fragmenttien ja aktiviteettien yhdistelyä.

Atte: Opin käyttämään gittiä, niin koneen sisäistä kuin githubiakin. Opin isomman kokonaisuuden hallintaa, varsinkin front-end puolelta, kuten monien fragmenttien yhdistämistä toisiinsa ja .xml tiedostojen tekemistä. Lisäksi opin ymmärtämään kielen toimintaa ja syntaksia paljon laajemmin, ettei kaikkea tarvitse etsiä googlesta.

Palaute harjoitustyöstä (vapaaehtoinen)

- Mitä toivoisit ensi vuoden harjoitustyöhön?

Vaikka applikaatio oli valinnanvarainen, pisteytyksen tarkat määritelmät rajasivat applikaation ominaisuuksia melko paljon (esim. pakollinen massan kuvaaja, joka ei liity muihin ominaisuuksiin). Toivoisimme siis ensi vuodelle enemmän valinnan varaa toteutukseen.