

HARJOITUSTYÖDOKUMENTAATIO

Ver. 1.0

Hyvinvointisovellus - **LiveGreen**

“Hyvinvointisovellus parempaan elämään”

<https://github.com/KeMaAid/OlioHarjoitustyo>

Team DPS:

Konsta Keski-Mattinen	0568752	konsta.keski-mattinen@student.lut.fi
Arttu Käyhkö	0569353	arttu.kayhko@student.lut.fi
Juha Kajanen	000200680	juha.kajanen@student.lut.fi

LiveGreen

“Hyvinvointisovellus parempaan elämään”

Sisällysluettelo

1. Johdanto	3
2. Kuvaus ohjelmasta	4
3. Käyttöohjeet	5
4. Tekijät ja työnjako	8
5. Ohjelman suunnittelu	9
6. Ohjelman toteutus	10
7. Luokkakaaviot	12
8. Toteutetut ominaisuudet	15
9. Työmäärät	16
10. Mitä opin harjoitustyöstä	18
11. Palaute harjoitustyöstä	19
12. Liitteet	22

1. Johdanto

Tässä dokumentissa kuvataan Team DPS:n harjoitustyön toteutuksen dokumentointi LUT-yliopiston CT60A2411 Olio-ohjelmointi -kurssia varten. Kurssin suoritus aika on kevät 2021. Harjoitustyöryhmän "Team DPS":n muodostavat seuraavat henkilöt:

Konsta Keski-Mattinen

Arttu Käyhkö

Juha Kajanen

2. Kuvaus ohjelmasta

LiveGreen sovellus on käyttäjilleen palvelu, jossa voi seurata ja verrata omien päästöjen aiheuttamista. Ohjelman tarkoituksena on herättää käyttäjän mielenkiinto kuluttamisen vaikutuksista ilmastoon, sekä tehdä päästöjen pienentämistä kilpailullista.

Käyttäjille luodaan tilit, joita tallennetaan tietokantaan. Tileillä on säilötty käyttäjästä tiettyä dataa ja salasanat ovat käsitelty kunnolla. Tietokanta haetaan sovellusta käynnistettäessä.

Sovelluksen pääikkunassa käyttäjä syöttää vetosäätimillä päivittäisen maito-, liha- ja kasvistuotteiden kulutuksen. Näistä lasketaan ulkoista rajapintaa apuna käyttäen käyttäjän hiilidioksidijalanjälki. Tuloksilla käyttäjä voi tehdä oman arvion nykyisistä kulutustottumuksistaan ja mahdollisten muutosten kautta vaikuttaa omalta osaltaan ilmastomuutokseen.

Dataikkunassa mittaustiedot näytetään graafisena esityksenä palkkikaaviona.

Sovelluksen kolmannella näytöllä käyttäjä voi tallentaa omia tietojaan kuten pituuden, painon, iän ja asuinpaikkansa. Pituuden ja painon avulla lasketaan käyttäjän BMI-arvo. Muita tietoja ei tässä vaiheessa käytetä hyväksi.

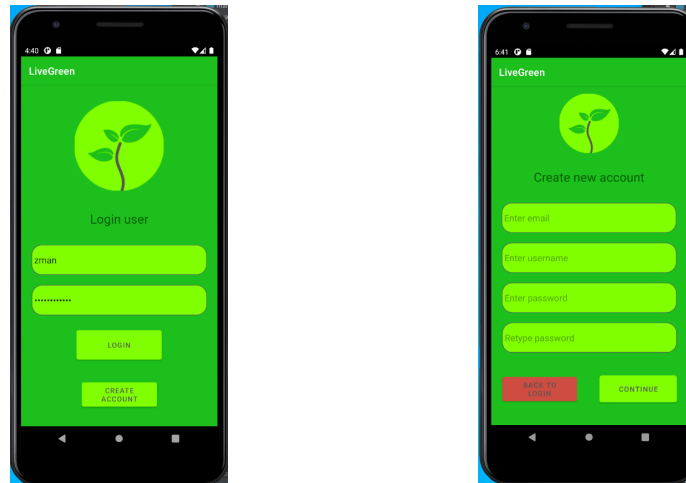
Profiiliikkunassa esitetään myös sovellukseen suunnitellun pelillistämisen perusajatus. Ikkunassa oleva "Ecorank" -kuvake kertoo käyttäjän ekologisista valinnoista pohjautuen hiilidioksidipäästöihin. Kuvake muuttuu käyttäjän kulutuksen perusteella. Tätä käytetään hyväksi eri käyttäjien väliseen vertailuun. Vertailu voidaan toteuttaa erilaisilla kaavioilla, alustavasti kahdessa eri tasossa, "ystävät" ja "kaikki käyttäjät". Kaavioista on mahdollista verrata itseään molempiin vertailutasoihin. Pelillistäminen toteutetaan siten, että pisteitä jaetaan käyttäjien sijoittumisesta kellokäyrille. Pelillistämistä tukee idea käyttäjien jakamisesta lohkoihin, jotka koostuvat tietyn osa-alueen suoriutumisesta. Annetut tiedot tallennettaisiin tietokantaan, josta ne luetaan sovellusta käynnistettäessä. Pelillistämistä ei sovellukseen vielä toteutettu, koska se vaatisi ulkoisen tietovaraston, johon kaikilla käyttäjillä on mahdollisuus kytkeytyä (pilvipalvelu).

Ohjelmaan syötetyt tiedot voi saada ulos formatoitua datana. Kirjoitus tiedostoon eri formaateissa (nyt CSV, jatkokehityksenä XML ja/tai JSON). Käyttäjän lokitietojen tallennus tapahtuu automaattisesti käyttäjän luonnin, sisäänkirjautumisen ja

uloskirjautumisen yhteydessä. Käyttäjän tallentamien tietojen uloskirjoitus tapahtuu oman painikkeen avulla.

3. Käyttöohjeet

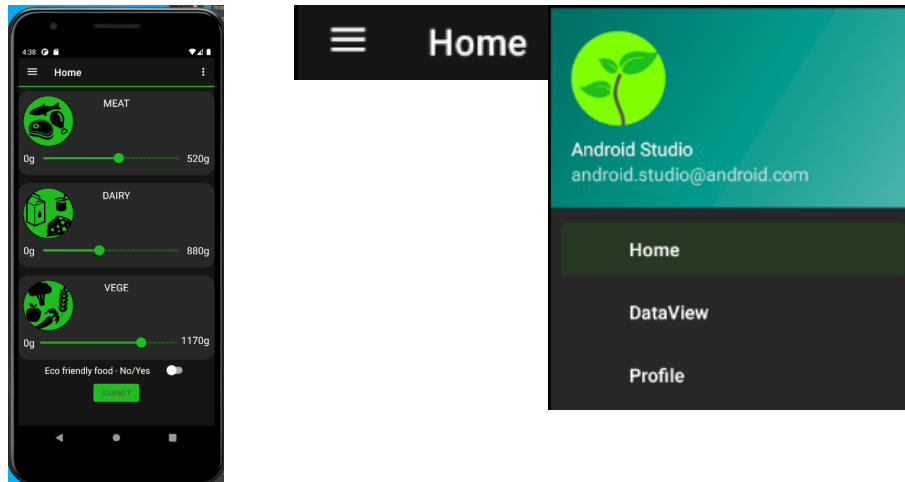
Ohjelma on hyvin yksinkertainen, sillä se on suunniteltu opastavan käyttäjänsä askel askeleelta. Ensimmäisellä käynnistyskerralla, tai jos käyttäjä on kirjautunut ulos, pääsee käyttäjä näkymään missä kysytään käyttäjätunnusta vai siirrytäänkö tilin luontiin. Tilin luonnissa käyttäjä vastaa yksiselitteisiin kysymyksiin luodessaan tiliä.



Kuva 1. Sisäänkirjautuminen ja käyttäjän luonti-näkymä

Käyttäjän luodessa tilin hän pääsee ohjelman koti-/datansyöttönäkymään. Samaan näkymään pääsee automaattisesti, jos käyttäjä on jo kirjautunut sisään ohjelmaa avattaessa.

Datansyöttönäkymässä käyttäjä voi täyttää tiedot muuta ohjelmaa varten. Vasemmalta löytyy menuvalikko hampurilaispainikkeen takaa. Sen avulla voi siirtyä muihin sovelluksen tarjoamiin näkymiin.



Kuva 2. Koti-/datansyöttönäkymä ja päävalikko

Kaavio-/Datanäkymässä näytetään käyttäjälle tiedot hänen antamistaan syötteistä pylväsdiagrammeina. Näitä diagrammeja on kaksi. Toisessa näytetään käyttäjän antamien syötteiden kokonaismäärä (grammaa per syöttö), ja toisessa puolestaan näytetään kuinka paljon kyseinen määrä ruokaa on tuottanut CO2 päästöjä (kiloina).

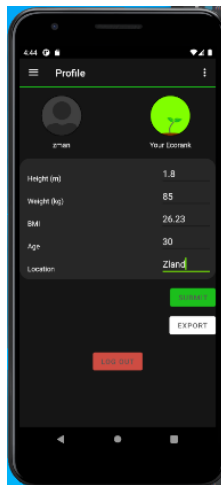


Kuva 3. Kaavio-/data-näkymä

Profiilinäkymässä näytetään vasemmalla ylhäällä käyttäjän oma profiilikuva ja hänen Ecorank-lohkokuvakkeensa, mikä kuvastaa käyttäjän normalisoitua päästöntuottoa.

Profiilinäkymässä käyttäjä voi antaa omat tietonsa ja sovellus laskee Submit-painiketta painettaessa niiden perusteella painoindeksin (BMI-arvon) käyttäjälle. Samalla käyttäjän tiedot tallennetaan tietokantaan.

Näkymässä on myös Export-painike, jolla käyttäjän tallentamat kulutustiedot kirjoitetaan tiedostoon, jota voi käyttää esim. taulukkolaskennassa. Näkymällä on myös sovelluksen Logout-painike.



Kuva 4. Profiili-näkymä

4. Tekijät ja työnjako

- Konsta Keski-Mattinen
 - UI/UX-suunnittelu
 - UI/UX-toteutus
 - Testaus
 - Login toiminnot
 - Dokumentaatio
- Arttu Käyhkö
 - Tietokannan login versio
 - Autentikointi
 - Bar Chart piirrot
 - Ulkoisen XML datan lukeminen
 - Laskulogiikka
 - Alustava login kirjoitus
 - Dokumentaatio
- Juha Kajanen
 - Järjestelmäarkkitehtuuri- ja tietokantasuunnittelu
 - Tietokannan ja tietokantakirjastojen toteutus
 - Alustakirjastot
 - Käyttöliittymä- ja alustakirjastojen integrointi
 - Järjestelmä- ja integraatiotestaus
 - Dokumentaatio

5. Ohjelman suunnittelu

Tarkemmin tietoa suunnittelupäätöksistä harjoistyön alussa palautetussa suunnitteludokumentissa ja alempana toteutettujen ominaisuuksien listassa.

Projektin suunnittelu toteutettiin noin viikoittaisissa, palaverissa jotka seurasivat kaavaa: Ideointi, kilpailukenttäanalyysi ja vaatimusmäärittely -> tarkempien rakenteiden valinta -> suunnitteludokumentin toteuttaminen. Ideoinnissa valitsimme mitä tulemme projektiksi tekemään ja yleisesti miten sen tulisi toimia. Kilpailukenttäanalyysi toteutettiin Google Play kaupan kautta, missä etsimme samankaltaisia sovelluksia, ja katselimme, mitä ratkaisuja kyseiset sovellukset olivat tehneet. Vaatimusmäärittely toteutettiin käyttäjäprofiilin luonnin avulla ja valitsemalla kohdekäyttäjälle tärkeät asiat.

Toteutettuihin valintoihin päädyimme siten ja siksi, että koimme kyseisten ominaisuuksien/komponenttien tarjoavan applikaatiollemme eniten. Jo työn alkupuolella teimme päätöksen yhdistää työhömmme Android Studion oman datan eli Room Persistence libraryn. Kannan kompleksisuus muuttui työn edetessä, sekä lopulta tarjolla on todella kattava kokonaisuus erilaisia syöttöjä, lukuja sekä toimintoja. Tauluja ohjelmassa on kaksi, jotka ovat Users ja Data. Ohjelman tarkoitus on olla oikeaoppisesti toteutettu kokonaisuus, joka tarjoaa helpon ja mielekkään jatkokehitysalustan. Toteutuksen muotista näkyvillä luokkakaaviot liitteissä alempana.

6. Ohjelman toteutus

Teknillisenä alustana toimii Android 4.1 - 11 käyttäen minimi sdk versiota 16. Ohjelmistossa on käytetty kirjastoja (alustavasti listattu alempaa löytyvän taulukon lopusta).

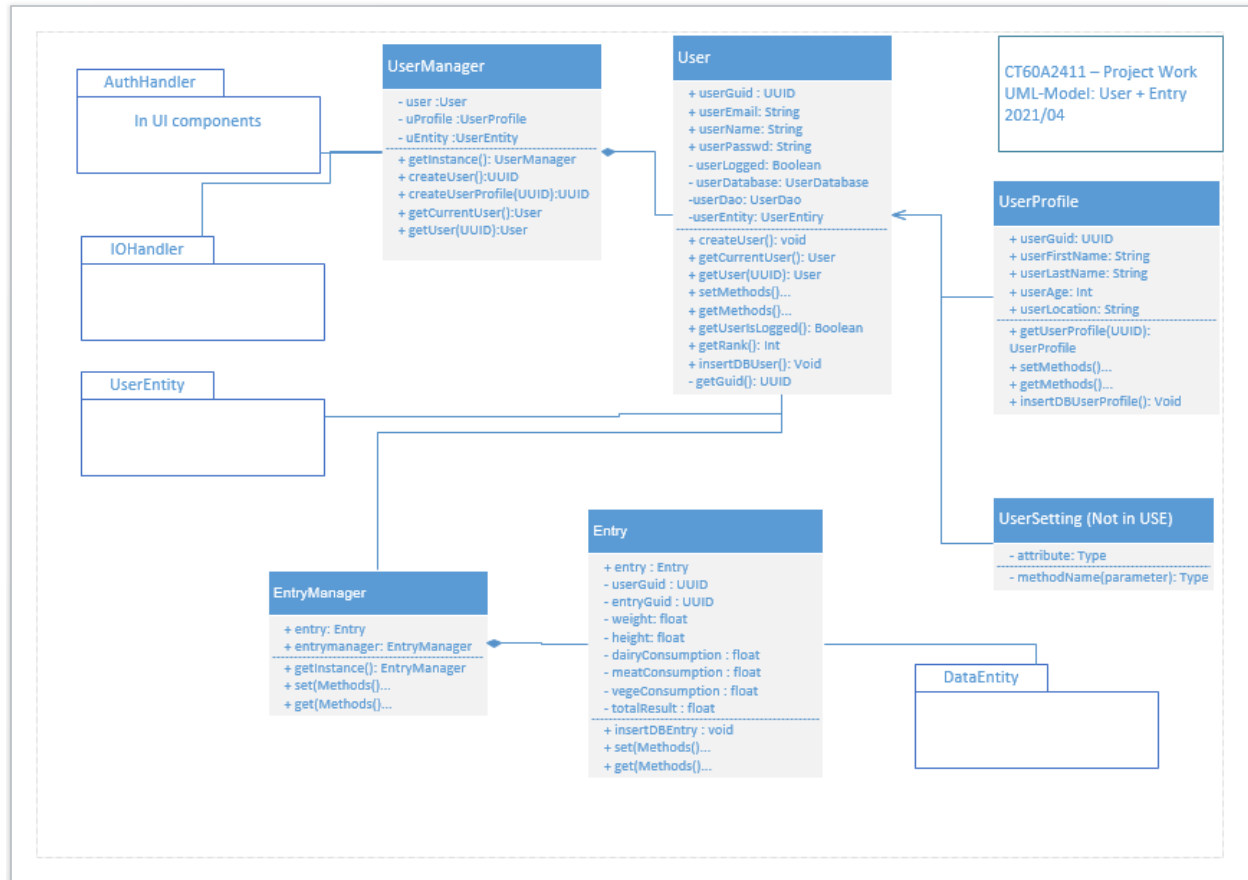
Taulukko 1: Käytettävät kehitys- ja työympäristöt

Tehtävä	Järjestelmä/ympäristö
Suunnittelu/Dokumentointi	Google Docs/Sheets/Slides
Suunnittelu/Aikataulutus	Quire
Digitaalinen prototyyppi	Figma
Data-arkkitehtuurisuunnittelu	Microsoft Visio
Ohjelmointi	Android Studio 4.1.2
Virtuaaliemulaattori	Pixel_3a_API_30_x86 / Android 11.0
Versionhallinta	Git/GitHub
Dokumentinhallinta	Git/GitHub
Tiimityö	Slack
Käytetyt nettisivut/lähteet	<p>Tietoa suomalaisten keskkulutuksesta: https://www.luke.fi/uutinen/mita-suomessa-syotiin-vuonna-2019/</p> <p>Keskivertosuomalaisen hiilijalanjälki: https://www.sitra.fi/artikkelit/keskivertosuomalaisen-hiilijalanjalki/</p> <p>SYKE Climate Diet public API 1.0 (Rajapinta kulutuksen aiheuttamien päästöjen laskentaan): https://ilmastodieetti.ymparisto.fi/ilmastodieetti/swagger/ui/index#/FoodCalculator</p>

Käytetyt kirjastot: MPAndroidChart library. Room persistence library. OkHttp.	Syy miksi käytetty: Kirjaston toiminnoilla BarChartin piirto. Kirjastolla datan basen toiminnot + autentikointi. Kirjastolla HTTP kutsu.
--	---

7. Luokkakaaviot

Sovelluksen tärkeimmät tietorakenteista ja sen käyttämät luokat esitetään kuvissa 1 - 2.

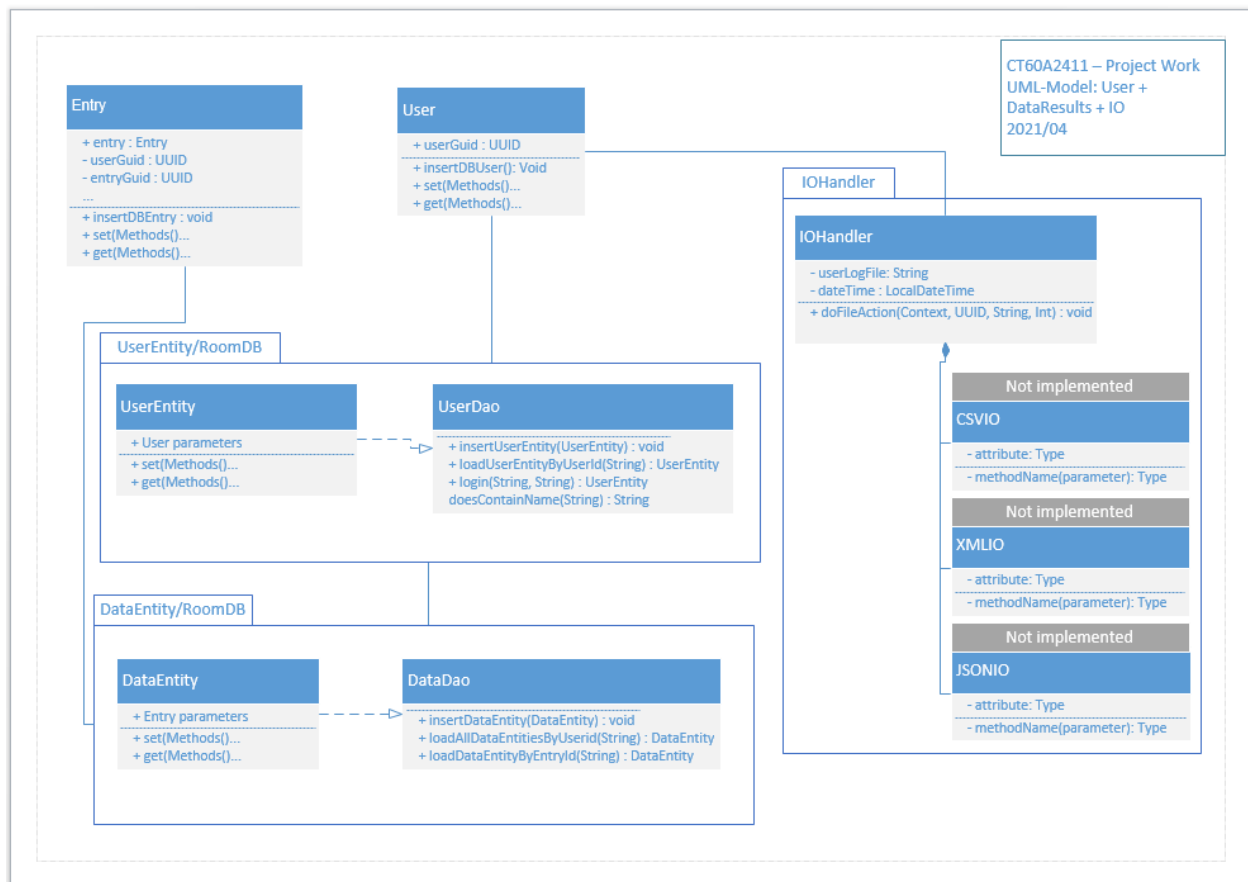


Kuva 1. Käyttäjä-entiteetit

Käyttäjän hallinta tapahtuu UserManager-luokan kautta. Käyttäjäluokka (User) jakaantuu kolmeen osaan, itse käyttäjäolioon ja siihen liittyviin aliluokkiin profiiliin (UserProfile) ja asetuksiin (UserSetting/ei toteutettu tässä vaiheessa). Profiiliin tallennetaan käyttäjän perustiedot (esim. nimi, asuinpaikka, käyttäjätunnus). Asetuksiin voidaan tallentaa toteutuksen jälkeen sovelluksen toimintaan liittyviä käyttäjäkohtaisia valintoja.

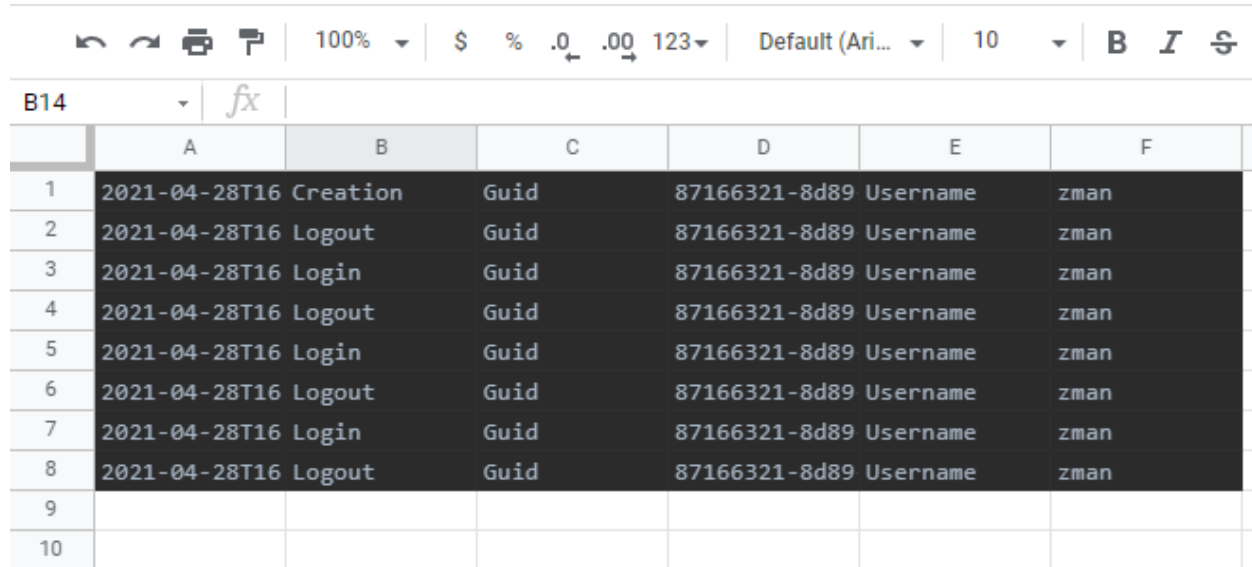
Käyttäjään liittyvät mittaustiedot hallitaan EntryManager-luokan kautta. Se toimii rajapintana mittaustietojen kirjaukseen, jotka tallentuvat Entry-luokkaan.

User- ja Entry-luokissa olevien tietojen tallennus tehdään pysyvään tietovarastoon, joka on toteutettu ulkoisen kirjaston avulla (RoomDB). Tietojen luku ja tallennus suoritetaan UserEntity ja DataEntity-luokkien välityksellä, jotka käyttävät User- ja DataDao-luokissa toteutettua tietokantarajapintaa.



Kuva 2. Käyttäjän tallentamien tietojen käsittely ja niiden uloskirjoitus

Lokitietojen tallennus tiedostoon tehdään IOHandler-luokkien avulla. Tässä vaiheessa on toteutettu vain .csv -muotoinen tiedostotallennus, jonka voi siirtää halutessaan esim. taulukkolaskennan käyttöön. IO-luokkia voidaan käyttää myös liittynöissä ulkopuolisiin tietovarastoihin mahdollisessa jatkokehityksessä.



	A	B	C	D	E	F
1	2021-04-28T16	Creation	Guid	87166321-8d89	Username	zman
2	2021-04-28T16	Logout	Guid	87166321-8d89	Username	zman
3	2021-04-28T16	Login	Guid	87166321-8d89	Username	zman
4	2021-04-28T16	Logout	Guid	87166321-8d89	Username	zman
5	2021-04-28T16	Login	Guid	87166321-8d89	Username	zman
6	2021-04-28T16	Logout	Guid	87166321-8d89	Username	zman
7	2021-04-28T16	Login	Guid	87166321-8d89	Username	zman
8	2021-04-28T16	Logout	Guid	87166321-8d89	Username	zman
9						
10						

Kuva 3. Lokitieto siirrettynä Google Sheet -ohjelmaan.

Käyttäjän syöttämien tietojen käyttö on toteutettu käyttöliittymäkomponenttien yhteyteen. Tietojen käyttö yhdessä ulkopuolisen datan kanssa on toteutettu samoin.

Käyttäjän autentikointi tapahtuu käyttöliittymässä ja sen alimetodeissa ja niihin liittyvät käyttäjätiedot (tunnus, salasana) tallennetaan User-luokkaa käyttäen.

8. Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Pakolliset	Pakolliset	13
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	Käyttäjäkokemus	2
Kirjautuminen applikaatioon	Käyttäjien eriyttäminen omille tunnuksilleen	3
Sovelluksella voi olla enemmän käyttäjiä	Käyttäjien eriyttäminen, jotta voivat kilpailla keskenään	3
Salasanavaatimukset	Kyberturvallisuus	2
Salasanan tallentaminen	Kyberturvallisuus	2
Ohjelma kerää päästöjen kehityksen ja näyttää muutokset graafisesti Bar Chart kaavioilla	Tarjotaan käyttäjälle visuaalinen tapa seurata hänen kulutustaan ja sen synnyttävää hiilijalanjälkeä	4
Asynkroninen kutsut ja multithreading	Mahdollistetaan datan käsittely keskeyttämättä käyttökokemusta, sekä tasoitetaan ajoa	3
Scoped storage käyttäminen	Pisteet ja harjoittelu	2
Fragmenttien hyödyntäminen aktiviteettien sijaan	Vetolaatikon kanssa toimiminen	2
Painon, pituuden ja iän syöttö + käyttö	BMI:n laskeminen profiilissa. Päivittyy muutoksien mukaan	4
Summa		40

Jatkokehitysideat:
Käyttäjäkuvakkeiden lisäämisen toteutus
Pelillistämisen toteuttaminen, käyttäjien välinen tietojen vertailu, tietokannan siirto pilvipalveluun
Käyttäjän kirjautumisstatuksen säilyminen myös sovelluksen sammuttua, ei vain jos ohjelma pysyy käynnissä taustalla (pysyy sisäänkirjautuneena, mikäli ei ole tehnyt uloskirjautumista).
Usean saman päivän aikana tapahtuneen datasyötteen yhdistys yhdeksi syötöksi.
Muiden ulkoisten datalähteiden kytkeminen ja niihin liittyvien uusien analyysien toteuttaminen
Laskenta- ja analyysitoiminnallisuuden kehittäminen

9. Työmäärät

Tekijä	Tehtävät	Tunnit
Keski-Mattinen	UI suunnittelu	10
Keski-Mattinen	UX	5
Keski-Mattinen	UI ohjelmointi	18
Keski-Mattinen	ikonografia	5
Keski-Mattinen	Fragmenttien tilajohdannaisuus	4
Keski-Mattinen	Erillisten komponenttien integrointi	3
Keski-Mattinen	Login logiikan uudelleenkirjoitus	10
Keski-Mattinen	Dokumentointi	2
Keski-Mattinen	Testaus	4

Keski-Mattinen	User, UserManager, UserProfile, userDao uudelleenkirjoitus	3
Keski-Mattinen	Salasanojen salaus	2
Keski-Mattinen	Yhteensä	66
Käyhkö	Käyttäjän luonnin, tallennuksen ja kirjautumisen pohja.	10
Käyhkö	UserDatabase (Room) alustava implementointi.	15
Käyhkö	BarChart piirrot.	7
Käyhkö	Ulkoisen sivun(XML) lukeminen ja parsiminen.	6
Käyhkö	Ulkoisten sivujen (XML) integrointi laskennan tarpeisiin (CO2).	8
Käyhkö	Pohja ulkoisten lokien kirjoitukselle.	4
Käyhkö	Datan siirto databaseen ja sen hakeminen.	3
Käyhkö	Yhteensä	53
Kajanen	Järjestelmäarkkitehtuuri- ja tietokantasuunnittelu	8
Kajanen	Tietokannan ja tietokantakirjastojen toteutus	12
Kajanen	Alustakirjastot	16
Kajanen	Käyttöliittymä- ja alustakirjastojen integrointi	20
Kajanen	Järjestelmä- ja integraatiotestaus	4
Kajanen	Dokumentaatio	4
Kajanen	Yhteensä	64
Summa		183 h

10. Mitä opin harjoitustyöstä

Keski Mattinen	Android studioin teeman ja visuaalisten komponenttien hallintaa. Syvensin omaa kokemusta gitin versionhallinnasta ja migraatiosta. DAO ja roomdb Android frameworkin hienouksia kuin backstack Ryhmytyötaitoja Splashscreen
Käyhkö	Olio-ohjelmointia. Javan toimintoja. Databasen käyttöä, toimintaa ja ominaisuuksia. Kirjastojen ja dokumentaation käyttöä. Git + GitHub käyttöä/versionhallintaa. Databasen migraatio ja version kasvatus. Tietojen/toimintojen hakeminen ulkoisesta lähteestä. Haettujen tietojen formatointi sopivaksi käyttötarkoitukseen. Ongelmanratkaisua ja internetin hyödyntämistä. Android Studioin UI komponentteja. Thread toiminnot. Lambda joissain konteksteissa.
Kajanen	Java-ohjelmointi yleisesti Lähteiden ja valmiiden komponenttien käyttö omassa koodissa Android Studioin perusteet Git/GitHubin käyttö RoomDB:n käyttö, tietokantarajapinnan rakentamisen kertaaminen

11. Palaute harjoitustyöstä

Ominaisuuksien toteuttamisesta	
Keski Mattinen	Hauska toteuttaa vapaata projektia, mutta tuntui, että projektin ei pahemmin saanut yhteistä säveltä.
Käyhkö	Ohjeistus olisi voinut olla hieman rajatumpi/riittävän määrittelevä. Kun oli "vapaat kädet", täytyi muiden tekemiä toteutuksia asiasta vähemmän tietävänä googletella jatkuvasti.
Kajanen	Harjoitustyön sisältö-/ominaisuusvaatimukset suht. balanssissa, vaikka edellä mainittu "vapaat kädet" pitää paikkansa.
Mikä oli syvin asia?	
Keski Mattinen	<p>Kun repoon ilmestyy ei integroitu komponentti asiasta, josta ajattelin pysyä erossa.</p> <p>Android layouttien toteutuksen kivikautisuus. Esimerkiksi pyöristettyyn tekstikenttään vaaditaan että itse askartelee pyöristetyn monikulmionsa.</p> <p>Oma taiteellinen kyvyttömyys</p>
Käyhkö	<p>Kurssin esitietojen vähyys. Koen, että kurssi olisi suotavaa käydä ainakin sivuaineen opiskelijana viimeisenä jos ollenkaan.</p> <p>Android Studio "ominaisuudet". Oli se sitten katoavat Toast messaget tai virtuaalinäppäimistön muutokset syystä x.</p>

Kajanen	<p>Android Studio. Oikeastiko tuolla epävakaaalla jumiutuvalla rautalankapurkkaviritelmällä joku tekee töitä. Tuottavuus lähenisi asympotoottisesti nollaa jos tällä pitäisi tehdä vakavasti töitä.</p> <p>(Muuten voi nostaa esiin etätyöskentelyn uusien ihmisten kanssa, mikä tietty asetti jonkin verran haasteita. Samoin ryhmän omat aikataulut töiden ja muiden opiskelujen osalta.)</p>
Mikä oli hyvää tässä työssä?	
Keski Mattinen	Sai oikeasti jotain ns oikea visuaalista aikaiseksi verrattuna muihin kursseihin.
Käyhkö	<p>Ryhmätyöskentely. En olisi itse ikinä kyennyt näin monimutkaiseen ja hienoon toteutukseen, mutta silti jaksettiin aina auttaa ja ohjeistaa, kun jokin helpohkokaan asia ei heti onnistunut.</p> <p>Kaikki mitä työstä oppi. Koen, että tämä kurssi vastasi ainakin kolmea alkeet kurssia (vähintään).</p>
Kajanen	Tavoitteena saada aikaan toimiva kokonaisuus itselle täysin uudella alustalla, uusilla työkaluilla ja uuden tiimin kanssa. Respektiä tiimikavereille.
Mitä toivoisit ensi vuoden harjoitustyöhön?	
Keski Mattinen	
Käyhkö	<p>Selkeämpää/tarkempaa ohjeistusta. Esimerkiksi kuinka käyttäjän luonti/tallennus tulisi toteuttaa.</p> <p>Avoimempia harjoitusryhmiä. Discord harjoitukset olivat mielestäni erinomaisia.</p>

	Vain harjoitustyöasioissa kului koko 15 minuuttinen aika.
Kajanen	Ainakin tuoreemmat opiskelijat olisivat luultavasti tyytyväisiä runsaammasta lähituesta (luennoista, harjoituksista) ja f2f-työskentelymahdollisuuksista. Menee Koronan piikkiin.

12. Liitteet

Liite 1. Harjoitustyön pakolliset- ja lisätoiminnallisuudet

Ominaisuus	Pisteet
Olio-ohjelmoitu	Pakollinen
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

Ominaisuus	Pisteet
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	1 – 5 pistettä
Kirjautuminen applikaatioon	3 pistettä
Sovelluksella voi olla useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin	3 pistettä
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä)	2 pistettä
Salasanan tallennus käyttää jonkinlaista hash-menetelmää ja suolausta (esim SHA-512 + salt)	2 pistettä
Jokin toinen datalähde fiksusti implementoituna (näitä löytää esim. https://www.avoindata.fi/ , https://www.europeandataportal.eu/fi tai https://thl.fi/fi/tilastot-ja-data/aineistot-ja-palvelut/avoin-data/avoimet-raiapiinnat) eli sovellus käyttää siis useampaa datalähdettä kerralla	2 – 5 pistettä
Ohjelmaan on mahdollista syöttää perustiedot (esim. pituus, paino, ikä/syntymävuosi), kuva, asuinkunta) käyttäjästä ja näitä arvoja käytetään jossakin	2 pistettä
Ohjelma kerää käyttäjän massan kehityksestä dataa ja näyttää muutokset graafisesti havainnollistaen ruudulla	3 pistettä
Ohjelma näyttää graafisesti ilmastodieetin tarjoamien arvojen muutokset käppyröillä (esim. kuinka lihan kulutus ja hiilijalanjälki on muuttunut aikojen saatossa)	3 pistettä
Ohjelma kertoo asuinkunnan, ikäluokan yms. riskitekijät (esim. kunnassa X tupakoitsijoita on Y%) pohjautuen THL:n dataan omien syötteiden lisäksi	2 pistettä

Ominaisuus	Pisteet
Ohjelma peilaa eri datalähteitä ristiin ja muodostaa uutta tietoa (THL:n dataa höystettynä tilastokeskuksen datalla ja saldona saadaan ulos kaavio Z)	3 pistettä
Ohjelma muistaa käynnistämisen / kirjautumisen jälkeen missä näkymässä käyttäjä oli ennen ohjelman sulkemista	2 pistettä
Asynkronisten HTTP-kutsujen käyttö dataa haettaessa	2 pistettä
Fragmenttien hyödyntäminen aktiviteettien sijasta käyttöliittymiä rakennettaessa	2 pistettä
Scoped storagen käyttäminen tiedon tallennuksessa (ei vaadi käyttäjän myöntämiä oikeuksia laitteen massamuistiin, vaan toimii omassa "hiekkalaatikossaan")	2 pistettä
Responsiivinen käyttöliittymä (toimii siis erikokoisilla ruuduilla sulavasti)	2 pistettä
Puutteellinen dokumentaatio	-1 – -5 pistettä
Ohjelmassa on ongelmia, jotka haittaavat sen käyttöä	-1 – -5 pistettä
Ohjelma ei noudata kunnollista oliomallia	-1 – -10 pistettä
Ohjelmakoodia on kirjoitettu/kommentoitu suomeks (huomaa, että käyttöliittymätekstit voivat tietysti olla suomeksi)	-3 pistettä
Ohjelmassa ja/tai dokumentaatiosta on rasismia, vihapuhetta tai muuta epäsoveliasta	-20 pistettä
Jokin oma hieno ominaisuus tai toiminto (tai useampi)	Max 5 pistettä per ominaisuus
Yhteensä (mikäli kaikki toteutetut ominaisuudet tuottavat enemmän kuin 40 voidaan lisäominaisuuksilla korvata huonompia, mutta yli 40 pistettä ei voi harjoitustyöstä saada)	Max 40 pistettä.

Liite 2. Dokumentaation sisältö:

- Lyhyt kuvaus ohjelmasta, jonka toteutitte
- Kuka teki, mitä teki (ryhmätyössä ei pelkästään nimiä etusivulla, vaan kirjoittakaa, kuka teki minkäkin osuuden)
- Miten ohjelma suunniteltiin ja toteutettiin
- Luokkakaavio (vähintäänkin tärkeimmät metodit ja attribuutit, mutta sellainen mistä saa selvää. Mahdollista olla erillinen kuva dokumentaation ohella, jos dokumentaatiossa kuva on epäselvä)
- Lista toteutetuista ominaisuuksista ja toiminnoista ja lopullinen pistemäärä näiden perusteella
- Jonkinlainen arvio käytetyistä tunteista per ryhmän jäsen (esim. 20pvä ja 5h per pvä = 100h)
- Jokaisen ryhmän jäsenen vastaus kysymykseen: "Mitä opin harjoitustyöstä?" (vastaus vähintään 3 virkettä) ja muita yleisiä kommentteja (jos sellaisia on)
- Bonus (ei pakollinen olla dokumentissa, mutta suotavaa): Mitkä ominaisuudet / toiminnot olivat helppoja / vaikeita toteuttaa. Tämä auttaa tulevaisuudessa kehittämään harjoitustyön ohjeistusta paremmaksi ja pisteytystä sopivammaksi
- Dokumentaation muoto on **PDF-tiedosto**