

# <LifeChanger>

Tekijät Jonas Koskula, Jouko Ruosteenoja ja Elias Tiilikainen

## Kuvaus ohjelmasta

Ohjelmamme on terveystietopalvelu, joka kannustaa käyttäjää elämänmuutokseen pienillä arkisilla asioilla. Ohjelma seuraa käyttäjän elämää perustuen hänen syöttämiinsä tietoihin ja kertoo käyttäjälle dataa näiden perusteella. Ohjelma kertoo esimerkiksi käyttäjän painoindeksin ja painon kehittymisen, sekä ruokailutottumusten mukaan käyttäjän aiheuttaman hiilijalanjäljen.

Ohjelma sisältää kirjautumisen omalla käyttäjätunnuksella ja salasanalla. Käyttäjän on myös mahdollista hallita omia perustietojaan, jotka hän syöttää sovellukseen. Kirjautuminen on sisällytetty, jotta sovellusta voi käyttää useampi käyttäjä samalla laitteella, eikä yksityisen käyttäjän terveystiedot ole näkyvissä mahdollisesti muille käyttäjille. Sovellus tallentaa käyttäjätiedot pilveen, joten omia tietoja voi tarkastella millä vain laitteella, josta sovellus löytyy. Sovellus vaatii käyttäjältä vahvan salasanan ja suojaa sen vielä ennen tallennusta.

Ohjelma sisältää perustietojen kysymisen, kuten käyttäjän iän, sukupuolen, pituuden ja painon. Näillä saadaan yleiskuva käyttäjästä ja näiden tietojen avulla saadaan laskettua käyttäjälle tietoa terveydestään ja antaa suosituksia hänelle. Pituuden ja painon avulla voidaan esimerkiksi laskea käyttäjän painoindeksi. Aiemmin syötetyt tiedot näytetään käyttäjälle ja hänen on mahdollista muokata tietojansa.

Ohjelma seuraa käyttäjän painon kehitystä laskemalla BMI:n sekä havainnollistamalla syötetyn painodatan kehitystä graafisesti. Mikäli käyttäjä muuttaa perustietojaan, niin painoindeksi lasketaan aina uusimpien syötteiden perusteella ja painograafi piirretään myös aiemmin syötetystä datasta.

Ohjelma kysyy käyttäjältä tämän kuluttamat ravintoarvot määrän ja laskee sen perusteella kokonaishiilijalanjäljen. Hiilijalanjälki-tulokset tallennetaan ja ne havainnollistetaan käyttäjälle graafisesti. Täten käyttäjä voi seurata oman hiilijalanjälkensä kehittymistä eri ruoka-arvoilla.

Ohjelma sisältää myös toiminnon, jossa kysytään käyttäjältä tupakkatuotteiden käytöstä ja ohjelma kannustaa käyttäjää lopettamaan tupakoinnin. Ohjelma kysyy päivämäärän viimeisimmästä tupakoinnista ja ohjelma laskee tupakoimattomat päivät, sekä lopettamisella

säästetyt rahat keskihinnan perusteella. Näillä kannustetaan käyttäjää parantamaan elämäntapojaan ja auttamaan elämänmuutoksen toteuttamisessa.

## Tekijät

Harjoitustyö toteutettiin ryhmätyönä Jonas Koskulan, Jouko Ruosteenajan sekä Elias Tiilikaisen toimesta. Harjoitustyö toteutettiin pääsääntöisesti yhteisesti ja työnjako oli tasaista. Koodasimme sovellusta yhdessä ja koko prosessin ajan keskustelimme laajasti yhdessä ohjelmaan sisällytettävistä ominaisuuksista, sekä kuinka teimme ne. Harjoitustyöryhmässämme ei ollut selkeitä rooleja eri henkilöillä, vaan kaikki osallistuivat jollakin tavalla jokaisen ominaisuuden luontiin. Pystyimme toteuttamaan lähes koko harjoitustyön yhteisesti samassa tilassa, eikä etäyhteyksiä tarvittu.

Alla olevassa taulukossa on listattu tärkeimmät ominaisuudet, sekä henkilöt, jotka olivat pääasiallisesti vastuussa näiden ominaisuuden luonnissa.

Ominaisuus:	Tekijä / tekijät:
Firestore luonti ja yhdistäminen	Jouko & Jonas
Kirjautuminen	Jouko & Jonas
Salasanan testaus	Elias
Salasanan hash	Elias
Henkilötietojen syöttö	Jouko, Jonas & Elias
Henkilötietojen haku databasesta	Jouko, Jonas & Elias
Henkilötietojen testaus	Elias
Painoindeksin laskeminen	Jouko, Jonas & Elias
Paino-graafi	Jouko, Jonas & Elias
Ruoka-arvojen syöttö	Jouko & Jonas
Ruoka-arvojen muuttaminen Apin vaatimiksi prosenteiksi	Jouko
Ruoka-arvojen testaus	Elias

Apin yhdistäminen sekä tietojen syöttö ja haku apista	Elias & Jouko
Kulutus-graafi	Jouko, Jonas & Elias
Tupakointi-laskuri	Jouko & Elias
Graafiset elementit (logot, värit yms.)	Elias
Luokka-kaavio	Jonas
Testaaminen ja toiminnallisuuksien korjaus	Jouko, Jonas & Elias

## Ohjelman toteutus

Ohjelma toteutettiin Java-ohjelmointikielellä AndroidStudiolla. Valmis ohjelma toimii kaikilla uusilla Android-puhelimilla, joissa on käytössä vähintään Android 4.1 (Jelly Bean).

Harjoitustyö on toteutettu hyödyntäen Googlen Firebase alustaa, jonne kaikki käyttäjän tiedot tallennetaan. Firebasen avulla varmennetaan käyttäjän kirjautuminen sekä siellä olevaan databaseen tallennetaan kaikki käyttäjältä kerätyt tiedot, jotka tallennetaan.

Harjoitustyön tallennuksessa hyödynsimme GitHubia ja varmuuskopioimme harjoitustyön vaiheita sekä GitHubiin, että OneDriveen.

Toteutuksessa hyödynsimme lukuisia AndroidStudion omia kirjastoja sekä graafien piirtämisessä jjo64 graphview kirjastoa. Myös Firebasen tuomia kirjastoja hyödynnettiin niin kirjautumisessa, kuin tietojen tallennuksessa. Monissa laskutoimituksissa hyödynsimme myös esimerkiksi Math-kirjaston työkaluja.

### Kuvaus käyttöliittymästä:

Sovellus avautuu kirjautumissivulle, josta voidaan ohjata myös luomaan käyttäjätili.

Kirjautuessa kirjautumistietoja verrataan databasesessa oleviin tietoihin ja jos käyttäjä on olemassa, hän pääsee kirjautumaan tililleen. Käyttäjä tilin luonnissa sovellus vaatii käyttäjää syöttämään sähköpostin sekä salasana kahdesti. Sovellus testaa, että salasana täyttää hyvän salasanan vaatimukset.

Kirjautumisen tai käyttäjätilin luomisen jälkeen sovellus avautuu päävalikkoon, jossa käyttäjää tervehditään ja eri painikkeilla käyttäjä voi siirtyä haluamaansa kohtaan sovellusta. Päävalikossa löytyy vaihtoehdot henkilötietojen syöttämiselle, ravintoarvojen syöttämiselle, painon tarkastelemiselle, tupakointilaskuri sekä uloskirjautumis painike.

User info -välilehdellä käyttäjä voi syöttää omat tietonsa. Sovellus kysyy käyttäjältä nimen, pituuden, painon, iän sekä sukupuolen ja syötön jälkeen tallentaa ne. Samalla välilehdellä käyttäjä voi myös muokata näitä arvoja. Mikäli käyttäjä on jo syöttänyt arvoja, niin viimeisimmät syötetyt arvot haetaan databasesesta ja ne syötetään kenttiin valmiiksi. Käyttäjän syöttämien tietojen oikeellisuus tarkistetaan.

Nutrition -välilehdellä käyttäjä syöttää ravintoarvonsa viikoittaisten arvojen perusteella ja sovellus laskee näiden arvojen perusteella aiheutetun kokonaishiilijalan jäljen Ilmastodieetin APIa hyödyntäen. Nämä arvot havainnollistetaan käyttäjälle graafisesti.

Weight -välilehdellä ohjelma laskee käyttäjälle hänen painoindeksinsä hänen syöttämien henkilötietojen perusteella ja ilmoittaa sen. Lisäksi ohjelma vertaa painoindeksiä yleisiin ohjearvoihin ja kertoo käyttäjälle hänen painoindeksin tason. Samalla välilehdellä käyttäjällä on myös mahdollisuus avata painograafi, joka havainnollistaa käyttäjälle hänen syöttämien painoarvojen kehityksen.

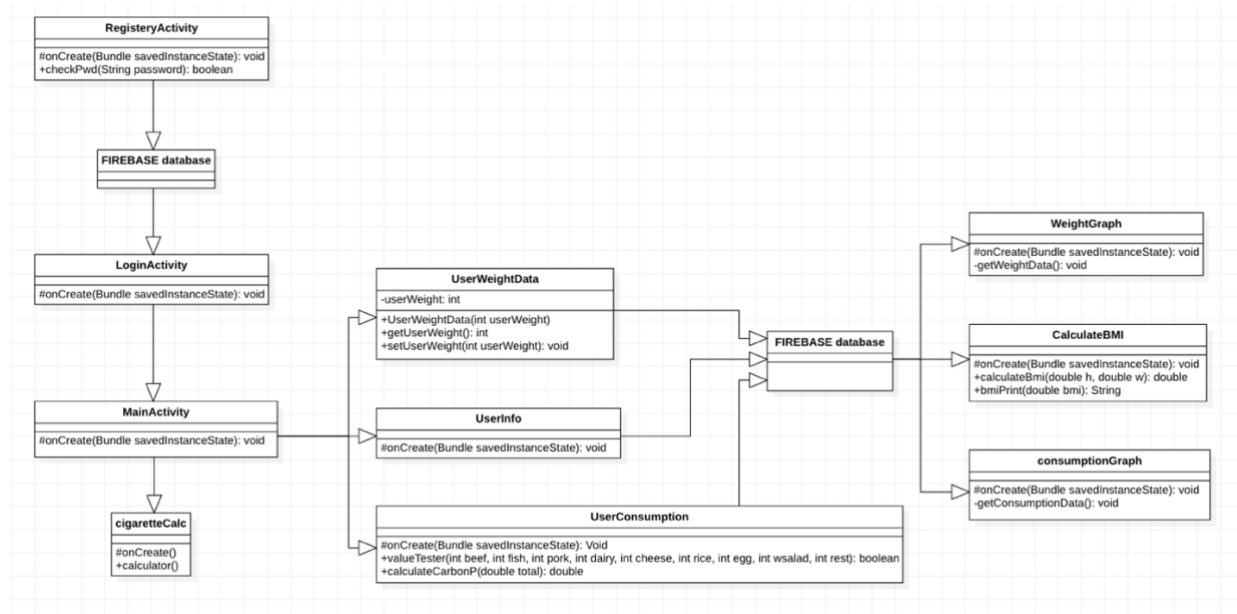
Smoking -välilehdellä käyttäjältä kysytään päivämäärä, jolloin hän on lopettanut tupakoinnin ja sovellus laskee erotuksen syötetyn päivän ja kyseisen päivän perusteella. Lisäksi sovellus kysyy päivittäin poltetun tupakkamäärän ja laskee tämän avulla tupakoinnin lopettamisella säästetyn rahan.

Log-out painikkeella sovellus kirjaa käyttäjän ulos.

Videolinkki sovelluksen esittelyvideoon: <https://youtu.be/dB7oLUxn6OI>

## Luokkakaavio:

Alla olevasta kuvasta näkyy luokkakaavio. Luokkakaavio löytyy myös erikseen liitteenä kuvana.



## Toteutetut ominaisuudet:

Ominaisuus	Perustelut	Pisteet
Olio-ohjelmoitu	Olio-ohjelmoitu	Pakollinen
Vähintään viisi erilaista luokkaa & oliota	Luokkia ja olioita on reilusti yli 5	Pakollinen
Vähintään yhden API:n käyttö, esim. Ilmastodieetti:	Ohjelma käyttää ilmastodieetti APIa	Pakollinen
Sovellus tallentaa käyttäjän toiminnan	Käyttäjän kirjautumistiedot ja henkilötiedot tallennetaan FireBaseen.	Pakollinen
Logia on mahdollista tarkastella eli voidaan tutkia arvojen kehitystä kirjausten edetessä	Käyttäjän painon kehityksen logia voidaan seurata kuvaajasta.	Pakollinen

Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	Ohjelma on selkeä ja värimaailma yhtenäinen. Tuo selkeyttä käyttäjälle, eikä sovelluksen käyttöönotto vaadi suuria it taitoja tai apua. Sovellus on tarkoitettu kaikille ja kuka tahansa voi ja osaa käyttää sitä.	5
Kirjautuminen applikaatioon	Applikaatioon pääsee sisään henkilökohtaisella sähköpostiosoitteella ja salasanaalla	3
Sovelluksella voi olla useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin	Applikaation eri käyttäjien rekisteröitymistiedot tallennetaan FireBase database järjestelmään. Käyttäjiä voi olla siis useampi.	3
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä.	Käyttäjän keksimän salasanan täytyy sisältää vähintään yksi numero, erikoismerkki, iso ja pieni kirjain ja olla vähintään 12 merkkiä pitkä.	2
Salasanan tallennus käyttää jonkinlaista hash-menetelmää ja suolausta (esim SHA-512 + salt)	Käyttäjän syöttämä salasana suojataan hash-menetelmällä tietoturvasyistä.	2
Ohjelmaan on mahdollista syöttää perustiedot käyttäjästä ja näitä arvoja käytetään jossakin	Ohjelmaan syötetään käyttäjän nimi, pituus, paino, sukupuoli ja ikä.  Painoa ja pituutta käytetään käyttäjän BMI:n laskemiseen.	2
Ohjelma kerää käyttäjän massan kehityksestä dataa ja näyttää muutokset graafisesti havainnollistaen ruudulla	Käyttäjän painon kehityksen logia voidaan seurata kuvaajasta.	3
Ohjelma näyttää graafisesti ilmastodieetin tarjoamien arvojen muutokset käppyröillä	Ohjelma tallentaa käyttäjän syöttämien tietojen perusteella kokonaishiilijalanjäljen ja havainnollistaa syötettyjen arvojen ilmastovaikutuksia graafisesti. Näin käyttäjä voi helposti seurata tottumuksiensa vaikutusta ja erilaisten arvojen aiheuttamaa muutosta.	3

Responsiivinen käyttöliittymä	Sovellus toimii käyttäjän omalla jo olemassa olevalla laitteella.	2
Käyttäjän datan tallennus pilveen.	Käyttäjän tallentamat tiedot eivät kuluta laitteen muistia	2
Kirjautumistietojen tallentaminen pilveen.	Sovellusta voi käyttää millä tahansa laitteella, eikä vain sillä, jolla käyttäjä on luotu	2
Tupakointilaskuri	Sovellus laskee käyttäjän syöttämän päivämäärän perusteella tupakoimatta olleet päivät ja säästetyn rahan.	2
Summa		44

## Työmäärät

Tekijä	Tehtävät	Tunnit
Jonas Koskula	Suunnittelu, devaus, testaus, dokumentaatio	70
Jouko Ruosteenoja	Suunnittelu, devaus, testaus, dokumentaatio	70
Elias Tiilikainen	Suunnittelu, devaus, testaus, dokumentaatio	70
Summa		210

## Mitä opin harjoitustyöstä?

**Jonas:** Harjoitustyön työstäminen oli todella opettava kokemus, joka kasvatti suuresti ymmärrystäni Olio-ohjelmoinnista ja miten se käytännössä tapahtuu. Opin harjoitustyöstä miten asiat sovelluksissa joita pidin itsestään selvyyksinä ovat välillä todellakin hankala toteuttaa. Tällaisia ovat esimerkiksi FireBase datan, sekä ulkoisen API:n käyttäminen sovelluksessa.



Harjoitustyön aikana opin myös varmuuskopioinnin tärkeyden, sillä Android Studion tapainen vaativa sovellus saattaa jumittua minä hetkenä hyvänsä.

**Jouko:** Harjoitustyö antoi hyvän näkökulman, millaista sovellusten tekeminen on. Opin käyttämään erilaisia valmiskirjastoja ja hyödyntämään monipuolisesti Android-Studion tarjoamia valmiskirjastoja. Harjoitustyön aikana ymmärsin esimerkiksi kommentoinnin, sekä varmuuskopioinnin merkityksen. Harjoitustyötä tehdessä huomasin, että Java ja android-studio ovat todella suosittuja työkaluja ja tietoa löytyi netistä todella paljon. Huomasin että olio-ohjelmoinnista alkoi tulla kivaa, sekä järkevää harjoitustyön edetessä. Tulevaisuudessa tulen varmasti käyttämään harjoitustyön aikana opittuja asioita ja mahdollisesti kehittämään osaamistani. Alkuun odotin harjoitustyön olevan todella haastava, mutta esimerkiksi Firebasen käyttäminen helpotti huomattavasti työskentelyä ja tietojen tallentaminen databaseen sujui mutkattomasti.

**Elias:** Harjoitustyö oli mielestäni hienosti rakennettu kokonaisuus, joka antoi laajan kuvan java-ohjelmoinnin mahdollisuuksista ja yleisesti sovelluskehityksestä. Itselläni oli melko hyvä pohja viikkotehtävistä, mutta harjoitustyö vaati silti todella paljon uutta tietoa. Opin harjoitustyön aikana, kuinka API toimii ja kuinka verkosta voi paremmin saada tietoa ja dataa sovellukseen. Laajan harjoitustyön toteuttaminen auttoi itseäni myös ymmärtämään, kuinka tarkkaa sovelluskehitys oikeasti on ja kuinka iso osa sitä kommentointi ja dokumentaatio on. Harjoitustyössä käytin myös ensimmäistä kertaa ulkopuolista palvelua tietojen säilymiseen ja olin itse yllättynyt kuinka hyvin Googlen Firebase soveltui tähän projektiin. Harjoitustyö vaati myös huomattavasti enemmän aikaa omalta osaltamme, kuin alkuun suunnittelimme. Harjoitustyön aikana oli kuitenkin jatkuvasti uusia haasteita ja ongelmia selvitettävänä ja näissä onnistuminen toi koko ajan lisää motivaatiota jatkaa työn tekoa.

## Palaute harjoitustyöstä

Ominaisuuksista meille yleisesti UI oli helppo toteuttaa ja yleisesti tietojen syöttäminen ja tallentaminen oli melko helppoa. Tässä meidän ryhmällämme kuitenkin Firebase oli iso tekijä ja verkosta löytyi kattavasti opetusvideoida tämän tueksi. Valmiiksi syötettyjen tietojen hakeminen kuitenkin aiheutti hieman päänvaivaa.

Harjoitustyön aluksi olisi voinut kuitenkin saada enemmän tukea ja vinkkejä työn aloittamiseen ja vaihtoehtoihin, kuten esimerkiksi mahdollisuus käyttää Firebasea tai SQL -kieltä datan hallinnassa.

API:sta olisi mielestämme voinut olla enemmän informaatiota. Jouduimme käyttämään paljon aikaa API:in aiheuttamien ongelmien ratkaisemiseen.

Aika-arvio oli mielestämme myös hieman optimistinen, me emme käyttäneet aikaa pienien asioiden "säättämiseen" ja muutenkin työskentelimme tehokkaasti. Silti aikaa kului paljon aika-arviota enemmän.