Kalorilaskin

Tekijät Lassi Laukkarinen, Akseli Hulkkonen ja Tommi Paakkunainen

# Kuvaus ohjelmasta

* Mitä ohjelma sisältää?

Ohjelman käyttäjä lisää päivittäisiä ruokamääriään sovellukseen. Sovelluksessa on kenttä, jonne käyttäjä arvioi päivä tai ateriakohtaisesti, mitä ruokatyyppejä käyttäjä on kuluttanut, (Nautaa, Kalaa, Possua, Meijerituotteita, Juustoa ja Salaattia).

Applikaatio hakee arvion syödyn ruoan päästöistä Ilmastodieetin ruoka-API:sta, ja samalla laskee arvion ruoan kalorimäärästä käyttäjälle. Näiden laskujen perusteella sovellus tallentaa silloin kirjautuneena olevalle käyttäjälle sekä päiväkohtaisen, että viikkokohtaisen kokonaiskalorinkulutuksen, sekä käyttäjän ruokailuista johtuneet hiilidioksidipäästöt viikkotasolla.

Lisäksi sovelluksessa on graafinäkymä, joka näyttää päiväkohtaisesti kulutetut kalorit, sekä hiilidioksidipäästöt kuluvan viikon ajalta. Sovellus tukee useita käyttäjiä, ja pitää erikseen kirjaa heidän kirjautumistiedoistaan sekä kulutetuista kaloreista ja päästöistä. Kirjautumisessa käyttäjän on luotava tili, jonka nimellä käyttäjän ravintotiedot tallennetaan erilliseen tiedostoon. Sovelluksessa voi olla useita tallennettuja käyttäjätilejä. Käyttäjätiliä luodessa käyttäjän on myös käytettävä salasanasuositusten mukaista salasanaa.

* Miksi nämä asiat on sisällytetty ohjelmaan?

Halusimme tehdä sovelluksesta helppokäyttöisen ja hyvin suunnitellun.   
Vaikka sovelluksen pääteema ja todennäköisesti suosituin ominaisuus on kalorilaskuri, halusimme tuoda sovellukseen mukaan myös hiilidioksidilaskurin. Tämä auttaa käyttäjää näkemään ruokailutottumustensa merkityksen ja kuinka paljon enemmän esimerkiksi sama grammamäärä lihaa tuottaa hiilidioksidipäästöjä kuin esimerkiksi kala-, tai kasvisvaihtoehdot.

Sovelluksen graafi auttaa käyttäjää myös näkemään ruokavalion vaikutuksen hiilidioksidipäästöihin helposti. Esimerkiksi käyttäjä voi nähdä syöneensä kolme päivää saman kalorimäärän ruokaa, mutta graafissa näkyvät hiilidioksidipäästöt vaihtelevat suuresti, riippuen siitä minkälaista ruokaa käyttäjä on syönyt. Tällä haluamme auttaa käyttäjää itse oppimaan, sekä pyrkimään kohti parempia ruokailutottumuksia itselle ja maapallolle. Sovellus tukee useita käyttäjiä siksi, että esimerkiksi lapsiperheen vanhempi voi kirjata myös lastensa ruoat erikseen, vaikka lapset eivät välttämättä pääsisikään käsiksi vanhemman puhelimeen.

# Tekijät

* Kuka teki ja mitä?

Lassi teki sovelluksen tietojen tallennusominaisuudet, sekä kirjautumisominaisuudet. Akseli teki sovelluksen käyttöliittymän, sovelluksen FoodActivityn toiminnallisuuksineen, sekä sovelluksen graafin. Tommi oli mukana sovelluksen suunnittelussa ja teki sovelluksen loppudokumentaation.

* Kuvaus työnjaosta ja rooleista

Niin sanottua roolitusta ryhmässämme ei ollut, käytimme projektin aikana Discord - sovellusta, jossa juttelimme projektin vaiheista ryhmämme kesken. Tommi keskittyi enemmän dokumentaatioon ja yleiseen ohjelman suunnitteluun, kun Akseli ja Lassi hoitivat itse konkreettisen koodaamisen.

# Ohjelman suunnittelu & toteutus

* Ohjelman suunnitteluprosessi

Suunnittelimme ohjelman yhteistyössä ryhmäläisten kanssa. Otimme etäpalaverin, jossa kävimme läpi määritellyt vaatimukset, ja valitsimme niistä ydinvaatimukset, jotka ainakin halusimme täyttää. Toki pakolliset, ja sen lisäksi koimme tärkeäksi erityisesti kirjautumisen, usean käyttäjän tukemisen ja tietojen tallentamisen. Ohjelmasta meille tuli mieleen kalorilaskurit, joihin käyttäjät voivat pitää kirjaa ruokailutottumuksistaan. Tähän pystyimme yhdistämään luontevasti myös ilmastodieetin tarjoaman päästölaskurin, jotta ohjelmalla voi seurata myös ruokailutottumuksiensa ympäristövaikutuksia.

* Millaisella teknisellä alustalla ohjelma toimii?

Sovellus toimii ja on testattu android – pohjaisilla mobiililaitteilla. Tätä pidemmälle emme lähteneet alustoja miettimään, kun käytössämme oli tällä aikataululla vain Android Studio. Kehityksessä ja testauksessa käytimme Andoid 11.0 (API 30) versiota. Aikaisemmilla versioilla saatetaan törmätä ongelmiin.

* Mitä kirjastoja on käytetty?

Käytimme standardien Android-kirjastojen lisäksi JSONReaderiä (org.json) ja MPAndroidChartia (com.github.mikephil.charting)

* Mitä työkaluja on käytetty?
  + Ryhmätyökalut

Discord – keskustelusovellusta kaikkeen harjoitustyöhön liittyvään kommunikointiin

* + Ohjelmistokehitystyökalut

Android Studio sekä GitHub versionhallinta

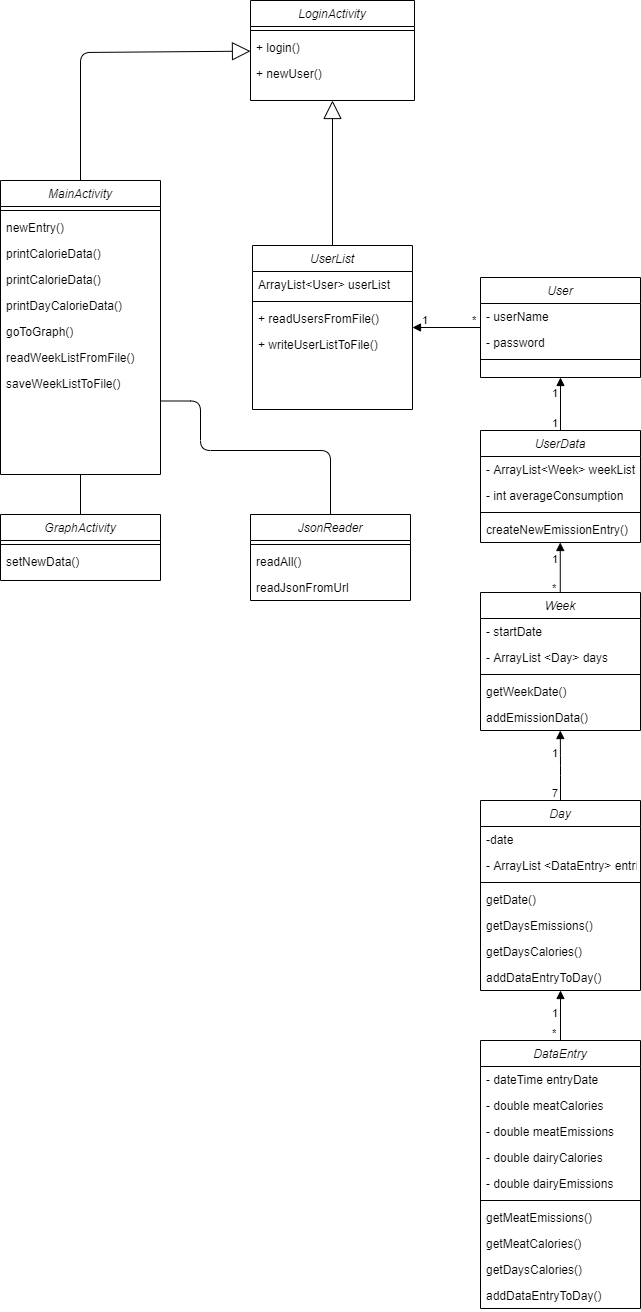
* + Testaustyökalut

Android Studion Virtual Device Manager

* + Dokumentointityökalut

Microsoft Word, Microsoft Sharepoint ja Drawiota UML-kaavioiden luomiseen.

# Luokkakaavio



# Toteutetut ominaisuudet

|  |  |  |
| --- | --- | --- |
| **Ominaisuus** | **Perustelut** | **Pisteet** |
| Olio-ohjelmoitu | Pakollinen | \* |
| Vähintään viisi erilaista luokkaa & oliota | Pakollinen | \* |
| Vähintään yhden API:n käyttö (Ilmastodieetti) | Pakollinen | \* |
| Sovellus tallentaa käyttäjän toiminnan (käyttäjän syöttämät arvot / tulokset) logiin (JSON) | Pakollinen | \* |
| Logia on mahdollista tarkastella (puhtaana tekstinä, graafisilla käppyröillä jne)  (Graafinen käyrä) | Pakollinen | \*  \*Yht 13 pistettä |
| Hyvin suunniteltu ja rakennettu | Suunnittelu ja huolellinen rakenne auttaa tekemään sovelluksesta käyttäjäystävällisemmän, sekä helpomman kehittää tulevaisuudessa | 3 pistettä |
| Kirjautuminen applikaatioon | Jotta sovellusta voi käyttää useampi, halusimme tehdä sovellukseen käyttäjädatan yksilöinnin. Jotkin käyttäjät voivat myös miettiä omia kulutustietojaan henkilökohtaisena, joten niihin pääsyn varmistaminen vain itselle voi olla käyttäjälle tärkeää. | 3 pistettä |
| Salasana noudattaa hyviä sääntöjä | Käyttäjän on tehtävä sovellusta käyttääkseen käyttäjätili ja sovellus tarkistaa käyttäjän salasanan. Salasanan on oltava väh. 12 merkkiä pitkä, sisältää väh. 1 numeron, erikoismerkin, sekä ison ja pienen kirjaimen. Sovellus ilmoittaa, jos käyttäjän salasana ei ole suositusten mukainen. | 2 pistettä |
| Sovellus muistaa käyttäjien tiedot | Sovellukseen luonteva piirre, myös liittyen kirjautumiseen ja käyttäjätileihin | 2 pistettä |
| Scoped Storage | Auttaa sovelluksen toimintaa ja käyttäjädatan tallennusta | 2 pistettä |
| Useampi käyttäjä | Tämä liittyy aikaisempaan kirjautumisfunktioon. Käyttäjällä voi esimerkiksi olla useita tilejä, esimerkiksi perheen vanhempi voisi halutessaan tehdä lapsilleen oman käyttäjän, tai kokonaisuuskäyttäjän perhekaloreiden laskuun. Ominaisuus tuntui muutenkin kalorilaskuriin luonnolliselta. | 3 pistettä |
| Summa |  | 28 pistettä |

# Työmäärät

|  |  |  |
| --- | --- | --- |
| **Tekijä** | **Tehtävät** | **Tunnit** |
| Tommi Paakkunainen | Projektin aiheen suunnittelu & projektin funktioiden suunnittelu ryhmän kanssa, Tiedonhaku sekä loppudokumentaatio ja videon tekeminen | 15 |
| Akseli Hulkkonen | Sovelluksen käyttöliittymän tekeminen, FoodActivity & toiminnallisuudet sekä graafi - ominaisuus | 25 |
| Lassi Laukkarinen | Sovelluksen tietojen tallennus, sekä kirjautumisominaisuudet, Testaus | 35 |
| Summa |  | 75 |

# 

# Mitä opin harjoitustyöstä?

Lassi: Suunnittelu on vaikeaa. Alkuun oli melko helppoa rakennella UML-kaavioita, mutta sitten kun projektiin kävi kiinni, kaikkea tarvittavaa emme todellakaan osanneet ottaa huomioon suunnitellessa. Tämä johti ohjelmoidessa siihen, että suunnitellusta rakenteesta poikettiin, ja lopullinen työ ei niin paljon seurannut alkuperäistä suunnitelmaa. Aikataulullisesti selvittiin juuri ja juuri, mutta yhdessä vaiheessa projektia teki mieli käytännössä aloittaa alusta ja tehdä jo valmiit ominaisuudet fiksummin. Siihen ei kuitenkaan ollut aikaa, vaan saimme lopulta korjattua työn sen hetkisen version. Tätä ennen en osannut käyttää GitHubia, ja toki Javalla koodaamistakin tuli harjoiteltua. Ehkä tämä oli opettavaisin kokemus ”oikeasta”, isommasta projektista, tähän asti olen tehnyt vain yksittäisiä konsolissa ajettavia pikku ohjelmia.

Akseli: Ajoissa kannattaa aloittaa sillä tämä ei ollutkaan mikään ihan pikkuinen projekti, jonka tekee yhdessä yössä. Tiimin välinen kommunikointi on tärkeää sillä, jos jonkin on epäselvää niin on tai ei ymmärrä jotakin toimintoa niin on helppo kysyä selvennystä/apua. GitHubin käyttäminen ja konfliktien selvittäminen on tuli tutuiksi projektin aikana. Tietenkin Javalla koodaaminen tuli tutuksi ja oli oikein mukavaa verrattuna kurssin alkuun, jolloin oli ihan pihalla, miten tämäkin nyt toimii.

Tommi: Android Studion käyttäminen ja varsinkin GitHubin käyttö oli itselle tärkeitä asioita, joita opin. Varsinkin versionhallinta. Olen itse hyvin kiinnostunut ohjelmiston suunnittelusta, joten alkusuunnitteluvaihe ja lopussa tämän dokumentaation tekeminen oli mielenkiintoista. Suunnittelu on tärkeää ohjelmoinnissa ja omasta mielestäni pärjäsimme hyvin, verratessa alkupään suunnittelua ja mitä sovelluksesta tuli nyt. Javassa minulla on vielä oppimista, mutta kuten python kurssinkin jälkeen, tulen varmasti jossain vaiheessa opettelemaan Javaa vielä lisää joidenkin pienien projektien kautta. Ryhmässämme kommunikaatio toimi hyvin. Isot propsit Akselille ja Lassille myös heidän tehokkuudestaan, jossa itselläni on vielä parantamisen varaa.

# 

# Palaute harjoitustyöstä (vapaaehtoinen)

Lassi:

Android Studion käyttäminen on tosi hyvä osa koko kurssia, sillä tähän asti ohjelmointikursseilla on tehty vain simppeleitä pikkuohjelmia, joita kukaan ei aja missään ”oikeassa” ympäristössä. Harjoitustyön tulos taas on ohjelma, joka oikeasti voisi olla ladattavissa Play-Storesta. (Jos joku sen jostain kumman syystä haluaisi käyttöönsä)

Kommentoisin vielä, että tehtävien kompleksisuus nousi aika hurjasti viimeisistä viikkotehtävistä harjoitustyöhön, ja sen takia tuntui vähän, että upottiin syvään päähän. Tavallaan olisi ollut kiva, jos Android Studion viikkotehtävät olisivat olleet osia yhdestä samasta ohjelmasta, ja lopussa niistä olisi kasvanut jo isompi kokonaisuus, joka toimii yhdessä. Toki tässä mallissa yhden viikon tehtävien missaaminen voisi olla melko katastrofaalista.

Akseli:

Pidin kovasti käyttöliittymän suunnittelusta ja toteutuksesta, nämä olivat sopivan helppoja ja tietyt osa-alueet olivat vähän hankalia. Vaikeinta oli tiedon siirtäminen aktiviteettien välillä mutta tästäkin selvittiin. Projektissa ei oikeastaan ollut mitään todella ikävää kohtaa missä olisi mennyt hermo.

Tommi:

Harjoituksista Android Studioon siirtyminen oli itselle positiivinen osa kurssia. Yhtäkkiä pääsikin tekemään asioita hiukan ”konkreettisemmassa” ympäristössä, eikä tehty enää vain tekstinluku – koodinpätkiä tai tekstipohjaisia valikko-ohjelmia. Oli mukava oppia paremmin näkemään sovellusten suunnitteluprosessia ja toteuttamaan isompaa kokonaisuutta.

Kurssin harjoitustehtävissä oli todella jyrkkä vaikeustason nousu. Itse vaihdoin myös viikkotehtävien tekemisen Eclipsestä Visual Studio Codeen, minkä käyttö tuntuu itselle ainakin helpommalta. Kurssin discordissa taisi olla muitakin, kellä oli hiukan ongelmia päästä Eclipsessä alkuun. Mutta tärkeimpinä kehityspisteinä itse näkisin kurssin vaikeustason nousun tasaamisen. Olen myös samaa mieltä Lassin kanssa. Itse opin ryhmätyön aikana eniten ohjelmiston suunnittelusta ja versionhallinnasta ja olisi hienoa päästä toteuttamaan projekti, jossa versionhallintaa voitaisiin viedä ihan loppuun asti juuri jonnekin sovelluskauppaan, josta muut opiskelijat voisivat mahdollisesti ladata projektin sovelluksen laitteilleen, ja sitä kautta sitten tekisimme ohjelmistokehitystä.