

Wellness Checker

Johannes Haapanen, Leevi Malinen ja Georgy Pokazeev

Kuvaus ohjelmasta

Lopullinen ohjelmamme poikkeaa hiukan alkuperäisistä suunnitelmista ja käyttöliittymäluonnoksista, osittain ehkä hyvällä ja osittain huonollakin tavalla. Ohjelman tarkoitus on kuitenkin suositella käyttäjälle eri liikuntamuotoja sen hetkisiin sääoloihin perustuen, sekä seurata käyttäjän liikkumisen määrää.

Sovellusta käynnistettäessä vastaan tulee ensimmäiseksi kirjautumisruutu. Jos käyttäjää ei ole luotu aikaisemmin voi käyttäjä siirtyä näkymään, jossa tämän voi luoda. Tämän jälkeen käyttäjä kirjautuu sovellukseen juuri luomillaan tunnuksilla.

Kirjautumisen jälkeen vastassa on kotinäkymä, jossa sovelluksen päätoiminnallisuus sijaitsee. Kotinäkymässä käyttäjä voi lisätä liikuntatunteja sen hetkiselle päivälle, joka tulee sitten näkyviin graafiin. Lisäksi näkymästä saa selville sen hetkisen päivämäärän ja säätilan. Säätilaa myös hyödynnetään suositusten antamisessa, jotka räätälöityvät kutakin säätilaa varten. Sää tiedot sovellus hakee Openweathermapista.

Kotinäkymässä graafin yläpuolella on myös viikonpäivänäkymä, joka on toteutettu Calendar- ja SimpleDateFormat-olioita käyttäen. Alkuperäisenä suunnitelmana oli, että käyttäjälle suunniteltaisiin koko viikon liikkumisaikataulu, johon tästä näkymästä olisi ollut hyötyä.

Käyttäjän kotinäkymässä syöttämät tiedot kirjoitetaan lokiin, joka sitten luetaan graafin piirtämiseksi. Lokissa voi olla vain 7 viikonpäivän arvot kerrallaan, minkä on tarkoitus pitää pylväsdiagrammi tarpeeksi selkeänä ja luettavana.

Kotinäkymän lisäksi sovelluksessa on asetus- ja tietonäkymät. Asetusnäköymästä voidaan vaihtaa paikkakuntaa, jonka sää tietoja sovellus hakee, sekä lämpötilayksikköä Celsiuksen ja Fahrenheitin välillä.

Tekijät

Lyhyesti Georgy teki pääasiassa käyttöliittymän rakentamiseen liittyviä asioita, kun Johannes ja Leevi keskittyivät pääasiassa itse ohjelmakoodiin. Johannes on tehnyt kuitenkin myös paljon asioita käyttöliittymän eteen ja Leevi jonkin verran.

Johannes toteutti ohjelmaan WeatherHandler –luokan, jossa kommunikointi api-rajapinnan kanssa oli toteutettu. Hän oli myös paljon mukana suunnittelemassa ja toteuttamassa sovelluksen käyttöliittymää Georgyn kanssa. Lisäksi luokkien yhdistäminen ja integrointi koko ohjelman toiminnallisuuteen oli Johanneksen toteuttama. Myös kirjautumissysteemi oli Johanneksen toteuttama.

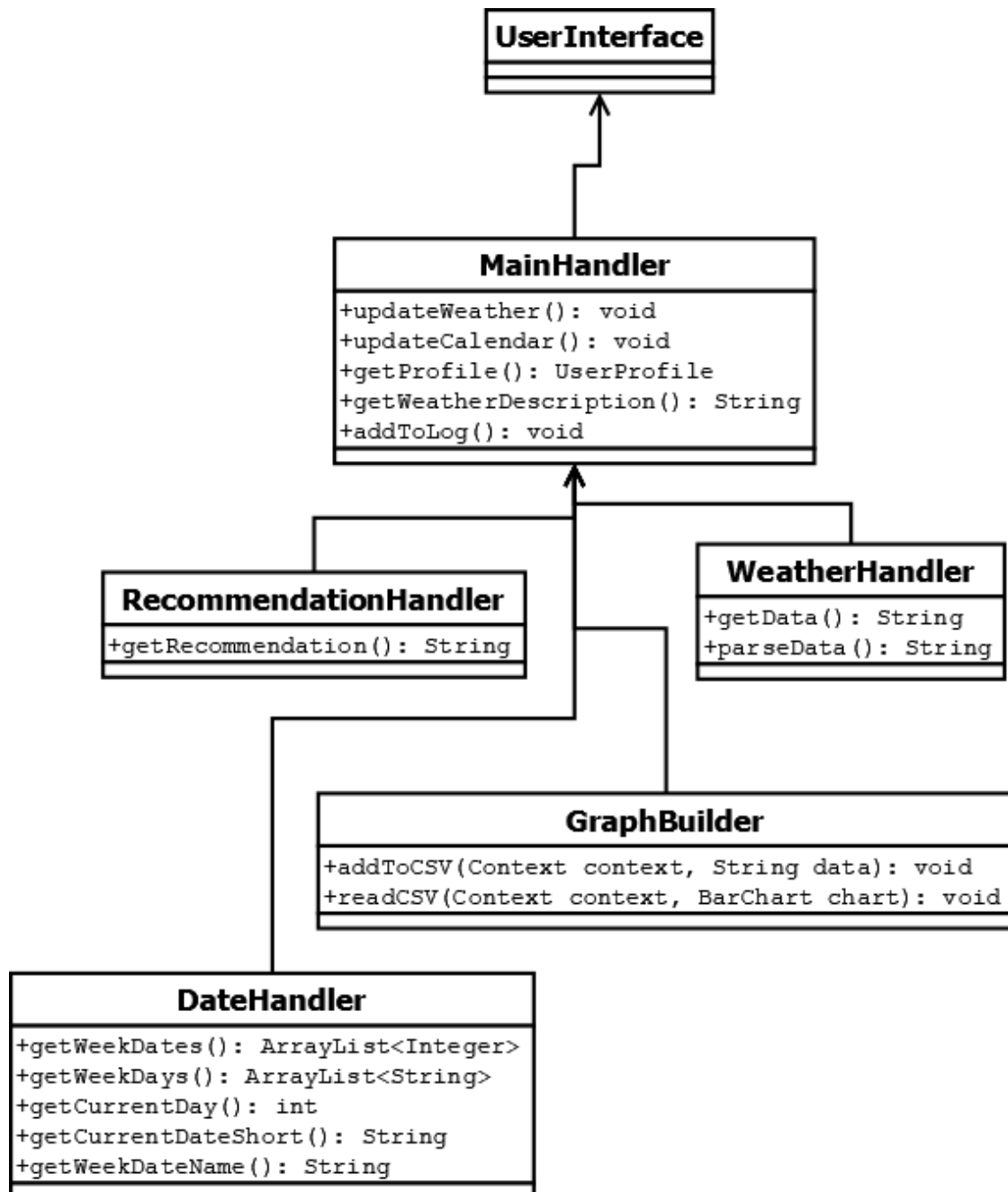
Leevi teki lähinnä käyttöliittymään liittyviä ohjelmointiasioita sekä pääasiassa GraphBuilder-luokan, jolle Johannes teki pohjatyön. Myös DateHandler, jonka pohjatyön Leevi ja Johannes tekivät yhdessä, on pääasiassa Leevin käsialaa. RecommendationHandler-luokka, joka arpoo päänäkymän lausahduksia, on suurimmaksi osaksi Leevin kirjoittama, vaikka Johannes auttelikin siellä täällä.

Georgy keskittyi pääasiassa kaikkien aktiviteettien käyttöliittymän suunnitteluun ja sen toteuttamiseen Android Studiolla. Suurin osa sovelluksessa käytetyistä ikoneista ovat myös hänen tekemiään.

Ohjelman toteutus

Ohjelma on kirjoitettu Android Studiolla, ja se toimiikin Android-alustalla. Ohjelmassa on käytetty Jackson-Databind ja MPAndroidChart -kirjastoja. Jackson-Databindista oli hyötyä API-datan käsittelyssä. MPAndroidChartin avulla taas saatiin havainnollistettua lokidataa kuvaajana. Versionhallintatyökaluna toimi Git ja koodin säilytyspaikkana toimi Github.

Luokkakaavio



Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Olio-ohjelmoitu	Pakollinen	Pakollinen
Vähintään viisi erilaista luokkaa	Pakollinen	Pakollinen
Vähintään yhden API:n käyttö	Pakollinen	Pakollinen
Sovellus tallentaa käyttäjän toiminnan	Pakollinen	Pakollinen
Logia on mahdollista tarkastella	Pakollinen	Pakollinen
Responsiivinen käyttöliittymä	Koimme, että sovellus vastaisi enemmän realimaailman ohjelmistoprojektia, jos responsiivisuus olisi toteutettuna.	2 pistettä
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	Jotta pystyimme toteuttamaan sovelluksen, täytyi meillä olla selkeä visio ja suunnitelma komponenteista, jotka halusimme toteuttaa.	1-5 pistettä
Kirjautuminen applikaatioon	Koska liikuntamäärät ovat henkilökohtaisia, halusimme että sovelluksessa olisi kirjautuminen	3 pistettä
Sovelluksella voi olla useampi käyttäjä	Kun kirjautuminen oli toteutettu, ei olisi ollut järkeä, jos sovelluksella ei voisi olla useampaa käyttäjää	2 pistettä
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä	Käyttäjiä täytyy pakottaa luomaan vahva salasana, koska muuten he eivät sitä tee	2 pistettä
Summa		23-28 p

Työmäärät

Tekijä	Tehtävät	Tunnit
Johannes Haapanen	Sovelluksen rakenne, api toteutus ja kaikkien osien toimiminen keskenään.	56
Leevi Malinen	Graafista, suosituksista sekä päivämääristä huolehtivat luokat, käyttöliittymän ohjelmointi	49
Georgy Pokazeev	Käyttöliittymän suunnittelu ja toteutus	47
Summa		152

Mitä opin harjoitustyöstä?

Leevi: Opin ainakin käyttämään gittiä paljon paremmin. Opin myös jonkun verran käyttöliittymän suunnittelemisesta Android Studiolla. Osaan nyt myös käyttää Javan Calendar ja SimpleDateFormat -olioita sekä MPAndroidChart-kirjastoa jollain tasolla.

Johannes: Opin lukemaan Googlen dokumentaatiota paljon paremmin. Projektin alussa täytyi tietoa hakea useista eri lähteistä mutta projektin edetessä ymmärrys Googlen dokumentaatiosta parani ja lopuksi riitti katsoa vain yhtä lähdettä, jotta suoritettava toiminnallisuus saatiin toimimaan. Lisäksi koin, että osasin projektin lopuksi selittää visioni ja ajatukseni paremmin käsillä olevasta työstä. Myös taitoni ohjeistamisessa paranivat.

Georgy: Opin todella paljon todella paljon käyttöliittymän suunnittelusta ja toteutuksesta. XML koodien tulkinnan ja kirjoittamisen lisäksi, olen tuottanut täysin itsetehtyjä ikoneita, joka on myös vaatinut oppimista. Ennen tämän kurssin harkkatyötä en ole osannut käyttää gittiä melkein ollenkaan, mutta nyt osaan käyttää sitä vapaasti.