

# Korona Apuri

Tekijä Elias Syyrilä

## Kuvaus ohjelmasta

- Mitä ohjelma sisältää?

Korona Apuri on koronatieosovellus, josta voi selvittää Suomen kaupunkien tartunta- ja rokotusluvut viikkotasolla. Kaupungin luvut esitetään haun kanssa samaan aikaan graafisesti pylväsdiagrammina, jossa x-akselilla on viikot alkaen 2020. Diagrammissa tartunnat ja rokotukset näkyvät päällekkäin.

Käyttäjä voi valita itselleen kolme suosikkikaupunkia, jotka tallentuvat heidän tililleen tietokantaa, ja esitetään aina kotinäkylässä päivitettyjen lukujen kera.

Saavutettavuuden parantamiseksi käyttöliittymän kielen voi vaihtaa suomen ja englannin välillä, ja käyttöliittymän värimaailman voi muuttaa yö-tilaan.

- Miksi nämä asiat on sisällytetty ohjelmaan?

Sovellus on pitkälti rakennettu tehtävänannossa ehdotetun THL data-muotin mukaisesti, ja esittää kyseiseen haasteeseen liittyvää informaatiota yksinkertaisessa ja järkevässä muodossa visualisointia hyödyntäen.

## Tekijät

- Kuka teki ja mitä?

Toteutin yksin projektin suunnittelun, devaamisen, debuggauksen ja raportoinnin.

- Kuvaus työnjaosta ja rooleista

Työn toteutettavat osa-alueet voidaan jakaa edellä mainittuihin tehtäviin, jotka suoritin pitkälti samassa järjestyksessä. Devaus ja debugaus tietenkin sekoittuivat samoissa työsesioissa.

## Ohjelman toteutus

- Millaisella teknisellä alustalla ohjelma toimii?

Ohjelma on rakennettu Android versiolle 7.0.

- Mitä kirjastoja on käytetty?

- MPAndroidChart-kuvaajakirjasto
  - <https://github.com/PhilJay/MPAndroidChart>
- SQLite
- JSON

- Mitä työkaluja on käytetty?

- Ryhmätyökalut

Github

- Ohjelmistokehitystyökalut

Android Studio Bumblebee | 2021.1.1 Patch 2

- Testaustyökalut

Android Studio Bumblebee | 2021.1.1 Patch 2

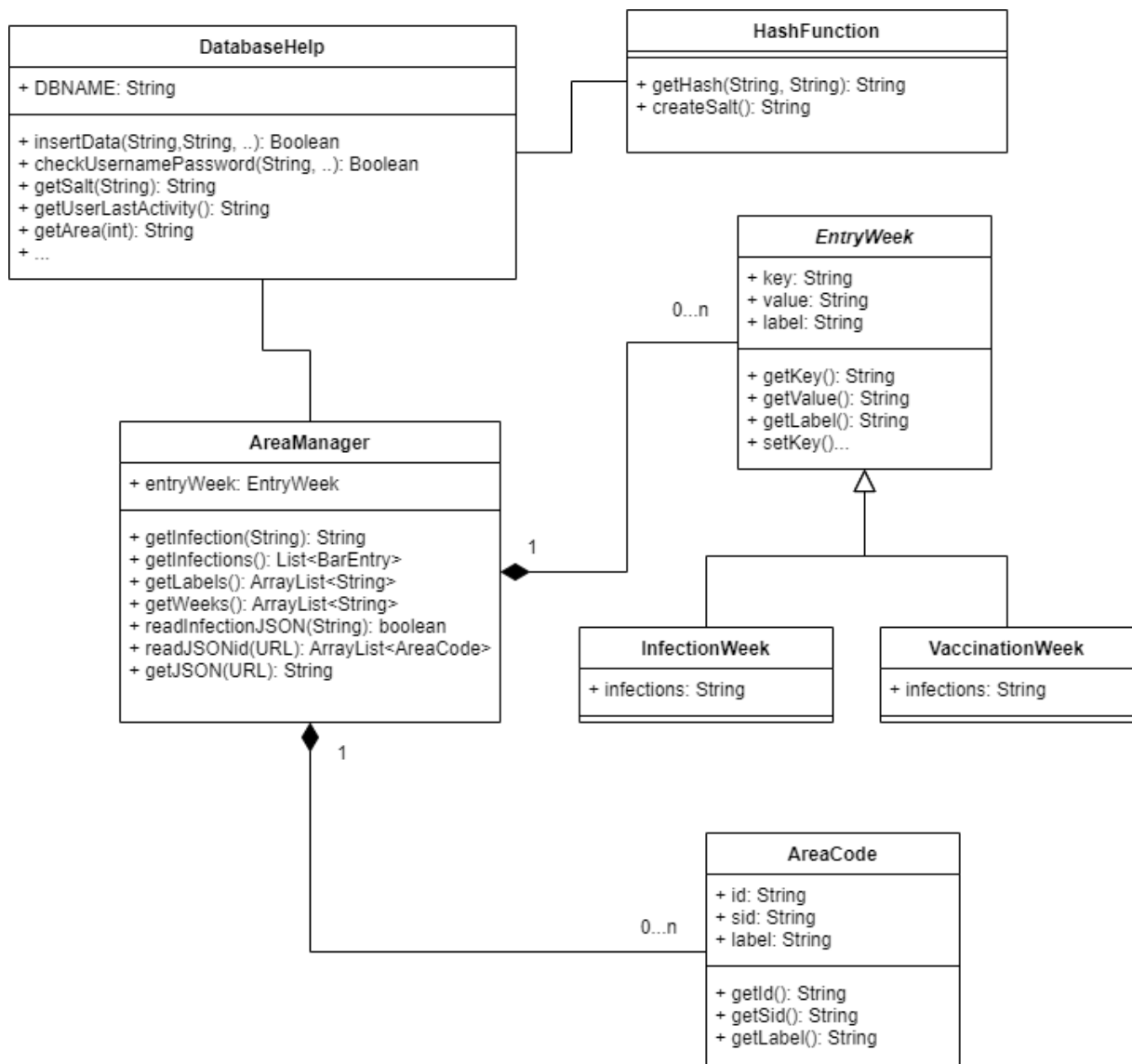
- Dokumentoitityökalut

Google Docs

Microsoft Word

Draw.io

## Luokkakaavio



## Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Perustaso	<ul style="list-style-type: none"> <li>- Ohjelman toiminta perustuu olio-ohjelmoinnin paradigmaan</li> <li>- Sisältää x luokkaa &amp; oliota, jotka eivät ole UI luokkia</li> <li>- Hyödyntää THL:n API:tä</li> <li>- Sovellus tallentaa käyttäjän toiminnan käyttäjän omaan kohtaan SQLite tietokantaan</li> <li>- Käyttäjän suosikkikaupungit, night mode-valinta, yms. säilyvät sovelluksen sulkemisen ja käynnistytksen välissä</li> </ul>	13

Hyvin suunnitellut UI komponentit	Sovelluksen jokaisessa näkymässä oleva sisältö on esitetty suunnitellulla tavalla hyödyntäen Androidin valmiita komponentteja. Jokaisen näkymän taustalla on jokin ajatus, jota kohti käyttäjää ohjataan otsikoilla, pudotusvalikolla ja napeilla, jotka reagoivat sopivalla tavalla käyttäjän syötteeseen. Pohjatasolla, rakenteen myös peilaavat muita moderneja mobiilikäyttöliittymiä. Värimaailman sekä päivä- ja yö modessa on räätälöity luomaan tietynlaisen teeman.	5
Kirjautuminen	Käyttäjä voi rekisteröidä käyttäjän syöttämällä käyttäjänimen ja salasanan kahdesti, ja kirjautua siihen myöhemmin.	3
Useampi käyttäjä	Käyttäjä voi luoda käyttäjiä rekisteröitymisnäkymästä. Jokaisen luodun käyttäjän tiedot tallennetaan SQLite-tietokanta muodossa. Erillisestä kirjautumisnäkymästä voi kirjautua kaikkiin tietokannan käyttäjiin myöhemmin myös sovelluksen uudelleenkäynnistyksessä.	3
Hyvän salasanan säännöt	Rekisteröitymisnäkymässä salasana-kenttä vaatii hyvän salasanan ominaisuudet, eikä anna luoda käyttäjää ennen kuin ehdot täyttyy.	2
Hash-menetelmä + suolaus	Rekisteröitymisen yhteydessä käyttäjän syöttämä salasana käytetään SHA-512 hash-funktion läpi muodostaen hash muodon. Hash-salasana tallennetaan SQLite tietokantaan, ja kirjautumisen yhteydessä käyttäjän syöte käytetään saman funktion läpi, ja verrataan tietokannassa olevaan. Rekisteröitymisen yhteydessä luodaan myös satunnainen suola, joka syötetään alkuperäisen salasanan kanssa hash-funktioon, ja tallennetaan myös sellaisenaan tietokantaan. Aina kirjautumisen yhteydessä, suola haetaan tietokannasta ja syötetään käyttäjän syötteen kanssa hash-funktioon.	2
Graafinen datan esitys	Haku-sivulla käyttäjän valitseman kaupungin korona tartunnat ja rokotukset näkyvät päällekkäin viikkotasolla pylväsdiagrammina.	4
Usealla kielellä	Asetus-sivulta käyttäjä pystyy muuttamaan pudotusvalikosta koko sovelluksen käyttöliittymän kielen joko suomeksi tai englanniksi.	2
Light mode/Dark mode	Asetus-sivulta löytyy myös napit, joilla käyttäjä voi vaihtaa koko sovelluksen käyttöliittymän elementtien teeman. Dark mode-asetus vähentää silmien rasitusta.	2
Ohjelma muistaa missä näkymässä jokainen käyttäjä oli ennen sulkemista	Ohjelma tallentaa viimeisimmän sivun käyttäjän omaan kohtaan SQLite tietokannan käyttäjien taulukossa.	2
Summa		38

## Työmäärät

Tekijä	Tehtävät	Tunnit
Elias Syyrilä	Suunnittelu, devaus, testaus, raportointi	60h
Summa		60h

## Mitä opin harjoitustyöstä?

Elias Syyrilä:

Opin...

- prosessoimaan json merkkijonoja JSONObject:eiksi, ja edelleen Java objekteiksi
- esittämään useampia tietolähteitä visuaalisesti Androidissa
- SHA-512 salasanasuojauksen ja suolauksen, sekä niiden toteutus käytännössä
- SQLite tietokannan käytön
- ja paljon Androidiin ja Javaan liittyviä temppuja

## Palaute harjoitustyöstä (vapaaehtoinen)

- Mitkä ominaisuudet / toiminnot olivat helppoja / vaikeita toteuttaa?

Rajapinnan json merkkijonon hankkiminen oli helppoa luennon pohjalta, mutta siitä eteenpäin ei ollut heti selvää, ettei käyttämäni lähde tuottanut jsonia, joka sopisi suoraan JSONArrayksi, joka oli luennon esimerkissä oletus. Tämä aiheutti turhaa ajanhukkaa, kun ei ymmärtänyt mistä oli kyse, kunnes aloitin GSONilla Java olioksi kokeilun, jolloin ajanhukka vasta alkoi.

Graafinen datan esitys käyttämälläni kirjastolla oli temppuileva ja monimutkainen matka, joka oli täynnä debuggausta ja kyyneleitä.

- Oliko jokin asia aivan syvältä?

GSON oli monimutkainen, ja lopulta mahdoton, soveltaa omaan tekemiseen, ja aikaa kului liian paljon sen kanssa yrittämiseen. Lopulta JSONObject menettely oli helppoa.

- Oliko jokin asia todella hyvää tässä työssä?

Avoimen datan rajapinnan käyttö oli mielenkiintoinen ja lopulta palkitseva kokemus.

- Mitä toivoisit ensi vuoden harjoitustyöhön?

Toivoisin, että ensi vuoden kurssilla olisi videota Android Studio debuggerin käytöstä ja breakpointeista, ja että JSONObject menettely json rajapinnan kanssa olisi mainittu heti ehdotuksena tehtävänannossa.