



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Oliver Martikainen, Katri Rasio,  
Christoffer Tverin, Jesse Väärälä

## Työpöytäsovellus opiskelijaravintoloiden löytämiseksi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotuotanto

Loppuraportti

1.5.2020

## Sisällys

1	Johdanto	1
2	Sovelluksen tilanne	2
2.1	Toteutetut toiminallisuudet	3
2.2	Toteuttamattomat toiminallisuudet	4
3	Arkkitehtuuri	6
3.1	Kuvaus arkkitehtuurista	6
3.2	UML-kaaviot	7
4	Testaus	9
4.1	JUnit5	10
4.2	TestFX	10
4.3	Manuaalitestaus	11
5	Käyttöohje	13
5.1	Käytön aloittaminen	13
5.2	Kartta	13
5.3	Sijainnin etsiminen	14
5.4	Ravintolalistan suodattaminen & ravintolan lisätiedot	14
5.5	Lähimmän ravintolan etsiminen	16
5.6	Omien asetusten määrittäminen	18
5.7	Ohjeet	19
	Lähteet	20

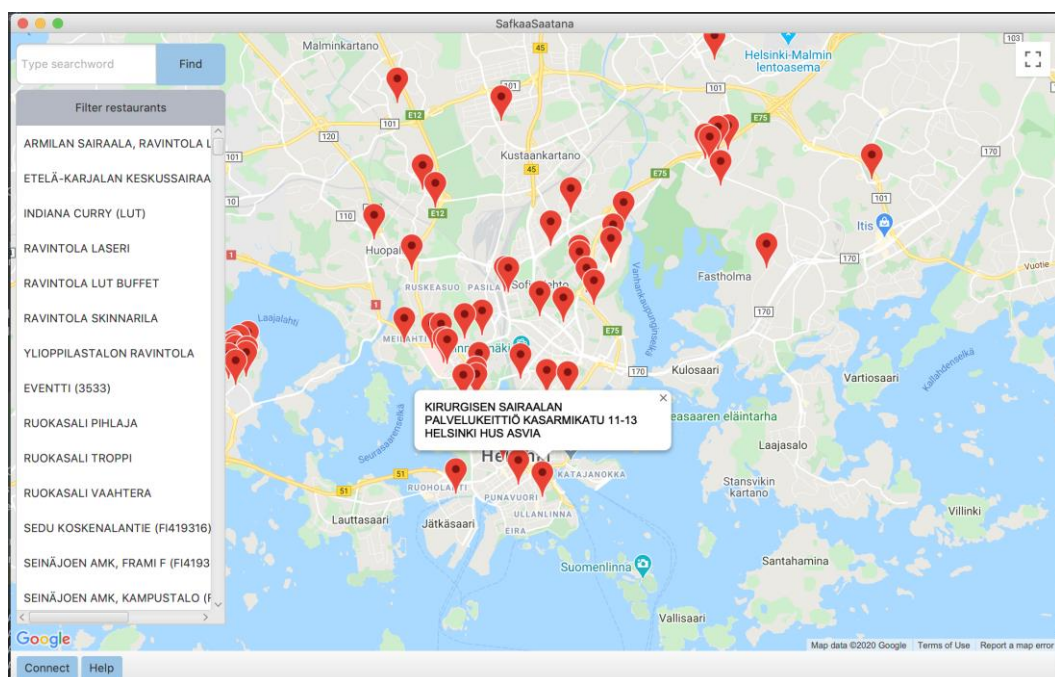
## 1 Johdanto

SafkaaSTNA –sovelluksella voi etsiä Kela-tuettuja opiskelijaravintoloita kartalla sekä reittejä niihin. Ohjelmaa käytetään JavaFX-työpöytäsovelluksena ja sen kohderyhmänä ovat opiskelijastatuksen omaavat henkilöt, jotka haluavat löytää ateriatuettuja ravintoloita ruokaillakseen edullisesti. Sovelluksen käyttö on maksutonta ja se pohjautuu vapaasti saatavilla olevaan aineistoon. Kilpailijoihin verrattuna sovelluksen etuna on se, että ravintolat näkyvät kartalla toisin kuin Kelan tarjoamassa opiskelijaravintolahaussa ja että käyttäjän ei tarvitse etukäteen määritellä kaupunkia, jonka alueelta ravintoloita haetaan.

## 2 Sovelluksen tilanne

Sovellus toteutetaan kahden periodin aikana. Tuotteen käyttäjätarinoista valittiin ensimmäiseen projektiin mukaan kriittisimmät tarinat, joiden toteuttaminen oli pakollista ennen kuin voidaan siirtyä toteuttamaan seuraavia.

Ensimmäisen projektin aikana toteutettiin ravintoladatan muodostaminen, tietokantayhteydet ravintoladataa varten, lista ravintoloista, karttanäkymä ja siihen piirtyvät ravintolamerkit, sekä osoitehaku Googlen karttaohjelmointirajapintaa hyödyntäen. Toteutimme sovellukseen myös pienimuotoiset ohjeet, kuinka sovellusta on tarkoitus käyttää.



Kuva 1: Sovelluksen varhaisemman version päänäkökulma

Toisessa jaksossa toteutimme sovellukseen kielivaihtoehdot, käyttäjän henkilökohtaiset asetukset ja niiden pysyvän tallentamisen sekä muutimme tietokantayhteyksiä, jotta sovellus olisi helpommin ja laajemmin käytettävissä. Kiinnitimme erityistä huomiota sovelluksen ulkoasuun ja helppokäyttöisyyteen.

## 2.1 Toteutetut toiminallisuudet

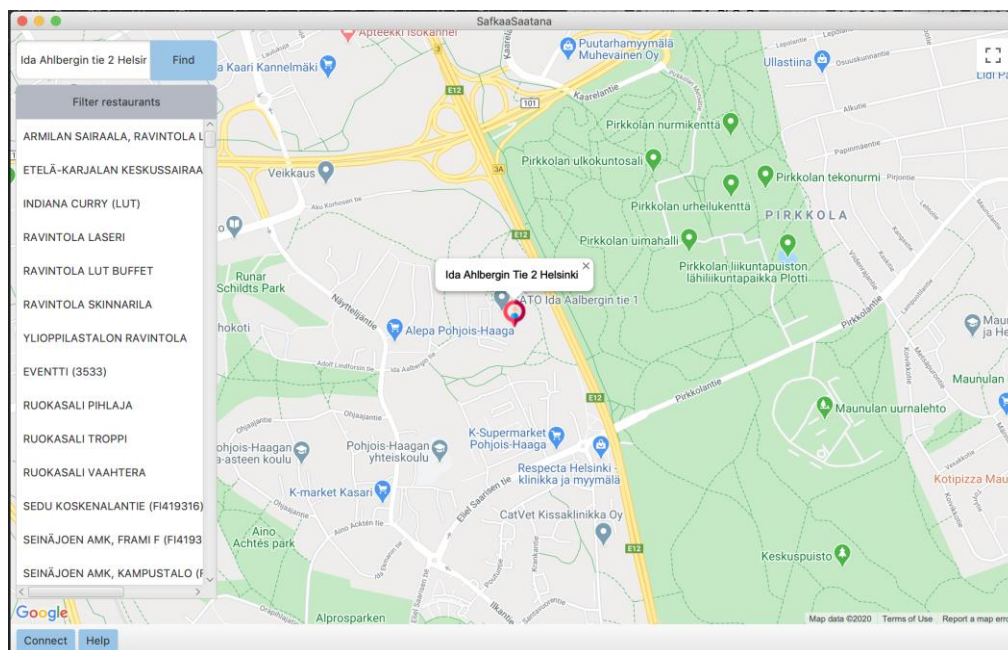
Sovelluksen pohjana on Kelan sivustolta yksinkertaisella Python-ohjelmalla JSON-tiedostoon koostettu data. Tämä tiedoston sisältämä tieto lisättiin sovelluksen varhaisella versiolla tietokantaan ravintoloina. Sovelluksen alkuperäinen SQL-tietokanta sekä Jenkins-automaatiopalvelin pyörii EDUcloud-palvelimella. Toisella periodilla päädyimme muokkaamaan tietokantayhteydet joustavampaan vaihtoehtoon ja siirsimme tietokannan MongoDB Atlas-alustalle. Muutoksen vuoksi sovellusta on mahdollista käyttää ilman Metropolian VPN-yhteyksiä, mikä mahdollistaa sovelluksen käyttäjäjoukon kasvun.

Molemmat tietokannat ovat yhä olemassa ja toimivat sovelluksessa yhteisen rajapinnan kautta; näin ollen tietokantayhteys on helppo vaihtaa tarvittaessa ohjelmakoodia kevyesti muokkaamalla. Tietokannan sisältämästä datasta toteutettiin myös ilman tietokantayhteyttä toimiva versio, sillä sovellus kirjoittaa ravintoladatan paikalliseen JSON-tiedostoon, josta ravintoladata käydään poimimassa, mikäli käyttäjä ei ole internetyhteyden äärellä.

Sovelluksen päänäköymänä toimii kartta, johon piirtyy automaattisesti sovelluksen avautuessa ravintoloita merkitsevät symbolit. Ravintolat ovat myös nähtävillä listalla ja listalta voi valita ravintolan, jolloin karttanäkymä keskittää näkymän valittuun ravintolaan ja avaa sivupalkkiin ravintolan lisätiedot. Mikäli ravintolan tiedoista löytyy ravintolan verkkosivuston osoite, on se avattavissa sivupalkissa olevasta linkistä.

Sovelluksella voi etsiä myös osoitteita. Osoitehaku toteutetaan Google Maps -ohjelmointirajapinnan kautta ja haun tulos osoitetaan kartalle asetetulla kuvakkeella. Sovellus etsii lähimmän opiskelijaravintolan ja keskittää karttanäkymän esittämään määritellyn lähtösijainnin ja sitä lähimmän ravintolan.

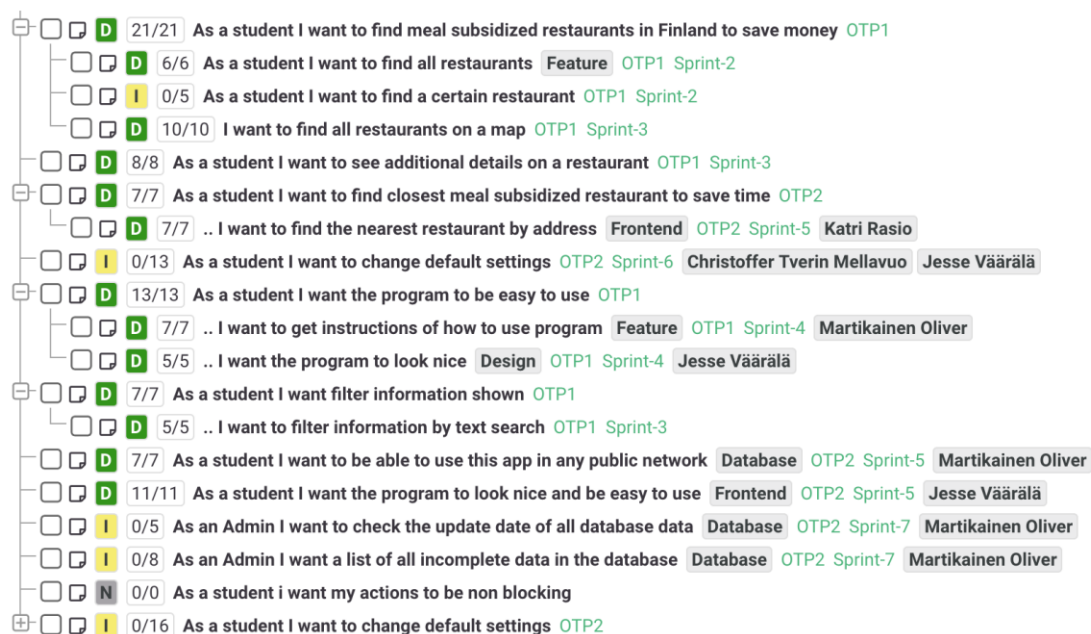
Sovellusta on mahdollista käyttää suomen, ruotsin ja englannin kielellä. Kielivalinta vaihtaa sovelluksen kaikki tekstit kohdekielelle välittömästi muutoksen tallentamisen jälkeen. Käyttäjän on mahdollista määritellä myös kaupunki, jonka kohdalle kartta keskitetään sovelluksen avautuessa. Sekä kielivalinta- että oletuskaupungin tiedot tallennetaan paikallisesti käyttäjän koneelle.



Kuva 2: Sovelluksen varhaisemman version osoitehaku

## 2.2 Toteuttamattomat toiminallisuudet

Projektisuunnittelun alkuvaiheessa käyttäjätarinat jaettiin kahden projektin aikana toteutettaviksi. Ensimmäisen projektin, OTP1:n aikana, saatiin toteutettua kaikki sille suunnitellut toiminnot ja ominaisuudet. Alun perin sovellukseen oli suunnitteilla reititys ja navigointiohjeet käyttäjän määrittämästä sijainnista lähimpään opiskelijaravintolaan, mutta tämä sovellus päätettiin karsia. Sovelluksen väriteeman muutosmahdollisuus jäi myös pois toteutuksesta.



Kuva 3: Tuotteen käyttäjätarinat

### 3 Arkkitehtuuri

#### 3.1 Kuvaus arkkitehtuurista

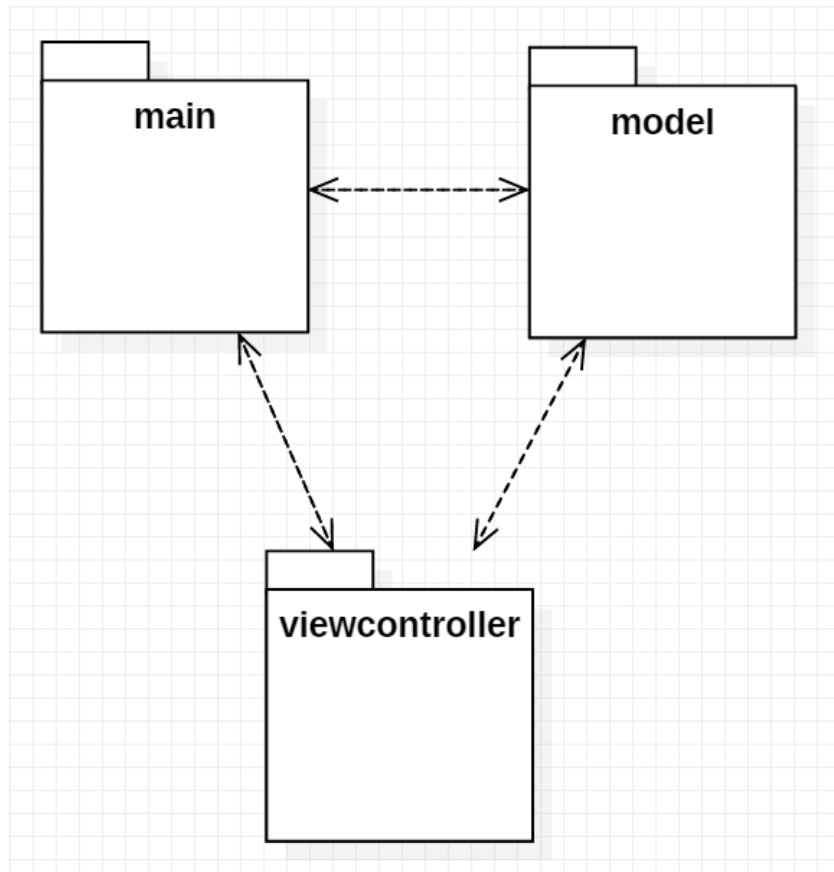
Sovelluksen arkkitehtuurin pohjana on muokattu MVC-malli. Mallikerroksella toimii data transfer object (DTO) -luokka, joka mallintaa tietokantaan tallennettua ravintolaa. Sovelluksen data access object (DAO) -luokkia on kaksi, sillä tietokantayhteyksiä on sekä SQL-tietokantaan EDUcloud-palvelimella ja noSQL-tietokantaan MongoDB Atlaksessa. Sovellukseen lisättiin myös mahdollisuus käyttää tietokannan tarjoamaa dataa ilman internetyhteyttä ja OfflineDatabase- ja OnlineDatabase-luokkien väliin on toteutettu Bridge-suunnittelumallin mukainen yhteys.

Sovelluksen näkymäkerrosta edustavat FXML-tiedostot, joilla kullakin on oma ohjain-luokka. Sovelluksen eri näkymien ja ohjainluokkien välinen kommunikaatio kulkee MainApp-luokan kautta. Tausta tälle ajatukselle pohjaa Mediator-malliin, vaikka toteutus ei vastaakaan Gamman mallia.

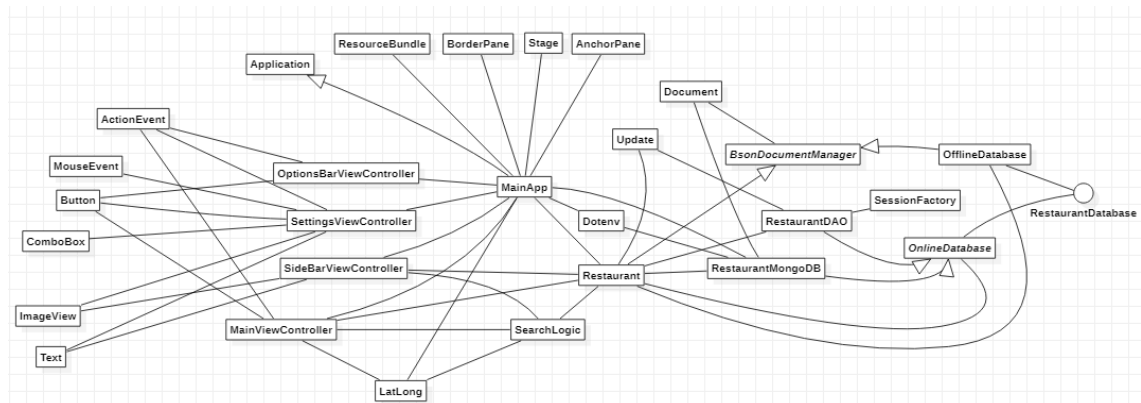
Sovelluksessa käytetään lukuisia kirjastoja. Karttatoiminallisuuden tarjoaa GMapsFX, graafisen käyttöliittymän JavaFX ja testauksessa käytetään JUnit5:sta, Surefirea, Jacocoa ja Mockitoa.



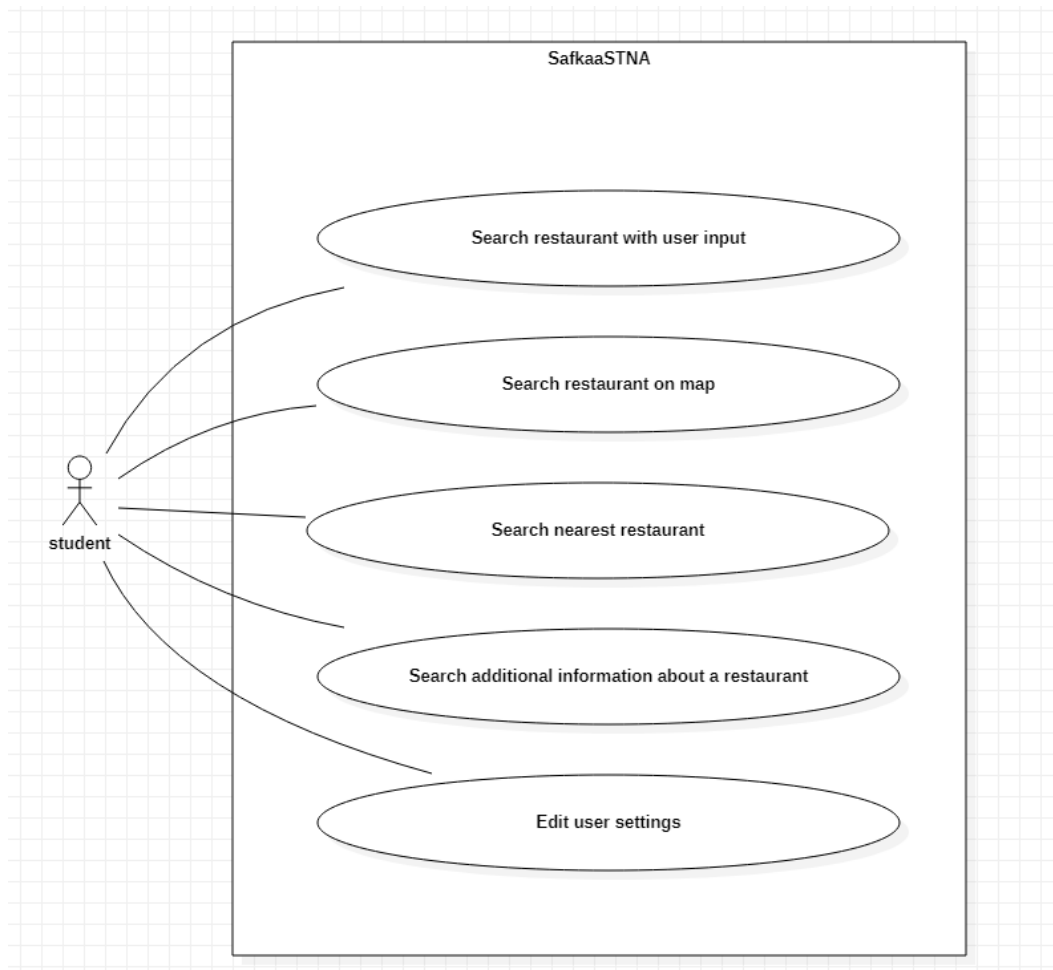




Kuva 5: Pakkauskäyttö



Kuva 6: Tyyppihierarkia



Kuva 7: Käyttötapauskaavio

## 4 Testaus

Sovellusta testattiin kehitystyön aikana runsaasti sekä automaattisesti että manuaalisesti. Sovellusta arvioitiin myös monia Käyttäjäkeskeinen suunnittelu -kurssilla opittuja malleja hyödyntäen.

Sovelluksen yksikkötestaus oli yksi projektin suurimmista haasteista. Sovelluksessa käytetty GMapsFX-kirjasto rajoitti testaamista rutkasti. GMapsFX-kirjaston kartan lataamiseen käytetty metodi jäi epäsäännöllisen säännöllisesti kirjastolta kutsumatta ja tämän vuoksi sovelluksen päänäköymän täyttämä kartta ei aina latautunut, mikä aiheuttaa testien epäonnistumisen ja sovelluksen kaatumisen. Edellä mainittu ongelma oli raportoitu kirjaston tiedetyissä ongelmissa, mutta emme runsaista yrityksistämme huolimatta saaneet ratkaistua sitä.

## 4.1 JUnit5

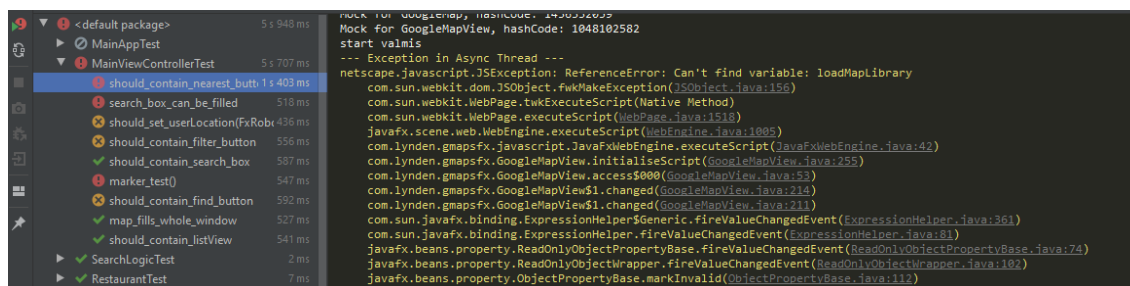
Sovellukselle kirjoitettiin JUnit5-testejä niiltä osin, kun se oli mahdollista. Tietokantaan liittyvät metodit testataan Mockito-kirjaston avulla, sillä sen avulla päästiin testaamaan tietokantaoperaatioiden toimintalogiikkaa ilman todellista yhteyttä tietokantaan. Yksikkötestin ei tule olla vastuussa muusta kuin sen testaamasta luokasta ja Mockitoa käyttämällä saimme rajattua esimerkiksi nettiyhteyden haasteet tai tietokannan mahdolliset katkot testin ulkopuolelle.

Sovelluksen ohjainluokkien metodit olivat myös haastavia testata, sillä suurimmalla osalla niistä ei ole palautusarvoa, vaan ne manipuloivat dataa, jonka perusteella karttaa muokataan. Karttasovelluksen komponenttien testaaminen ei onnistunut edes Mockiton avulla, sillä vaikka testattavan komponentin simuloisi Mockiton avulla, muodosti kirjaston silti yhteyksiä Google Maps-rajapintaan ja näiden kutsujen viiveiden vuoksi ensimmäinen ajettavista testeistä epäonnistui aina, sillä kartta ei ollut valmis ensimmäiseen testiin siirryttäessä.

Monet karttaan edes löyhästi liittyvistä JUnit5-testeistä toimivat lokaalisti yksittäisinä JUnit-testeinä, mutta Mavenilla ajettaessa ne epäonnistuivat lähes poikkeuksetta kartan latausongelmien vuoksi.

## 4.2 TestFX

Yritimme testata sovellusta myös TestFX-testikirjaston avulla, mutta samat haasteet GMapsFX-kutsujen kanssa aiheuttivat yhtä lailla ongelmia, jotka aiheuttivat testien epäonnistumisen.



Kuva 8: GMapsFX-kirjaston kartan alustuksen ongelmat testatessa

### 4.3 Manuaalitestaus

Sovellusta testattiin kehitystyön aikana manuaalisesti niiltä osin, mitä ei yksikkötesteillä saatu suoritetuksi. Alla olevaan taulukkoon on kirjattu tapaukset, jotka testattiin manuaalisesti.

Kategoria	Testin numero	Testitapaus	Oletus	Toteutus
Suodatustoiminto	1	Ravintolalistan rajaaminen hakusanalla 'unicafe'	Lista päivittyy näyttämään ravintolat, joiden nimessä, osoitteessa tai kaupungissa esiintyy termi 'unicafe'	ok (Kuva 11)
Suodatustoiminto	2	Ravintolalistan suodatuksen päälle ja pois päältä laittaminen	Filter-painikkeen teksti muuttuu, Search-painike passivoituu, ravintolalistan sisältö päivittyy	ok (Kuva 11)
Osoitehaku	3	Osoitehaku hakusanalla 'narinkkatori'	Kartta tarkentaa kohteeseen 'narinkkatori'	ok
Lähimmän ravintolan etsiminen	4	Etsitään lähin ravintolan kohteesta 'narinkkatori'	Sovellus löytää lähimmän ravintolan, avaa sen lisätietonäkymän ja keskittää kartan niin, että sekä Narinkkatori, että lähin ravintola näkyvät kartalla. Lähimmän ravintolan	ok

			kuvake on erilainen kuin muiden ravintoloiden.	
Lähimmän ravintolan etsiminen	5	Etsitään lähin ravintola ilman että lähtöpaikkaa on määritelty	Sovellus kysyy lähtöpaikkaa ja toteuttaa sen jälkeen kohdan 4.	ok
Asetusten muuttaminen	6	Kielen muuttaminen	Kaikki sovellukset tekstit muuttuvat valitulle kielelle	ok (Taulukko 2)
Asetusten muuttaminen	7	Oletuskaupungin muuttaminen	Sovellus keskittää karttanäkymän oletuskaupunkiin sovelluksen avautuessa	ok

Taulukko 1: Manuaalitestaus testitapauksilla

Kohde	Näkymä	suomen kieli	ruotsin kieli	englannin kieli
Etsi	Päänäkymä	Etsi	Sök	Search
Ohje	Päänäkymä	Ohje	Guide	Help
Asetukset	Päänäkymä	Asetukset	Inställningar	Settings
Etsi lähin	Päänäkymä	Etsi lähin	Hitta närmaste	Find nearest
Suodatus päällä	Päänäkymä	Suodatus päällä	Filter aktivt	Filter is active
Suodatus poissa	Päänäkymä	Suodatus poissa	Filter frånkopplad	Filter is inactive
ENG	Asetukset	ENG	ENG	ENG
SV	Asetukset	SV	SV	SV
FI	Asetukset	FI	FI	FI
Päivitä	Asetukset	Päivitä	Uppdatera	Refresh
Tallenna	Asetukset	Tallenna	Spara	Save

Taulukko 2: Kielivaihtoehtojen manuaalinen läpikäynti

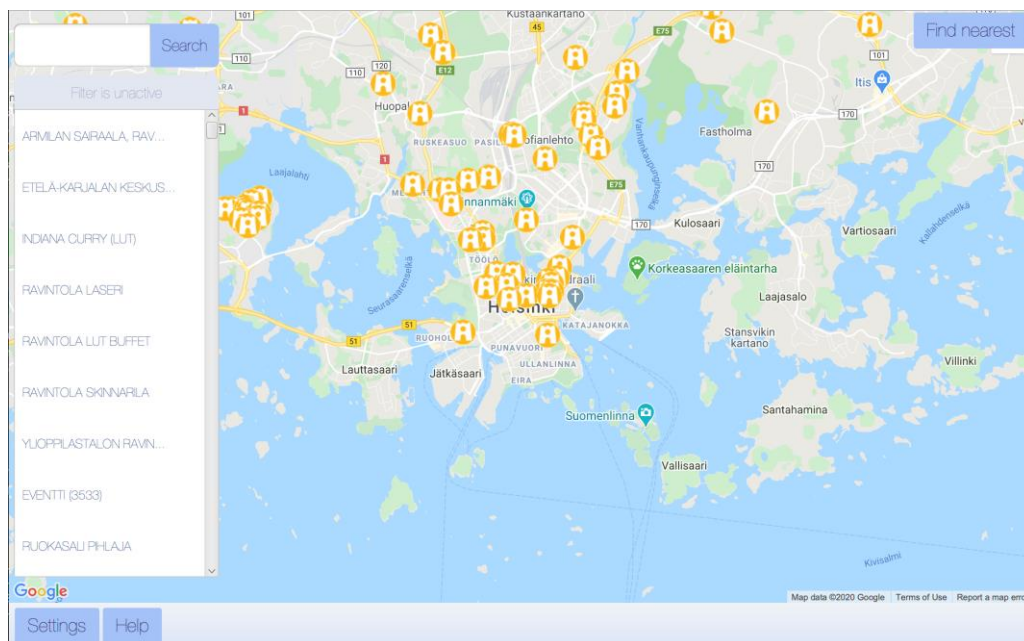
## 5 Käyttöohje

Sovelluksen käyttäminen on pyritty tekemään mahdollisimman yksinkertaiseksi.

### 5.1 Käytön aloittaminen

Sovellus avautuu ensimmäisellä käynnistyskerralla englannin kielellä ja kartta keskittyy Helsingin seudulle. Sovelluksen vasemmasta yläkulmasta löytyy kaksitoiminen hakukenttä. Hakukentän alapuolelta löytyy lista Kelan ateriatuetuista ravintoloista.

Oikeassa yläkulmassa sijaitsee painike, jolla voi etsiä lähimmän ravintolan ja alapalkista pääsee käsiksi asetuksiin ja ohjeisiin.



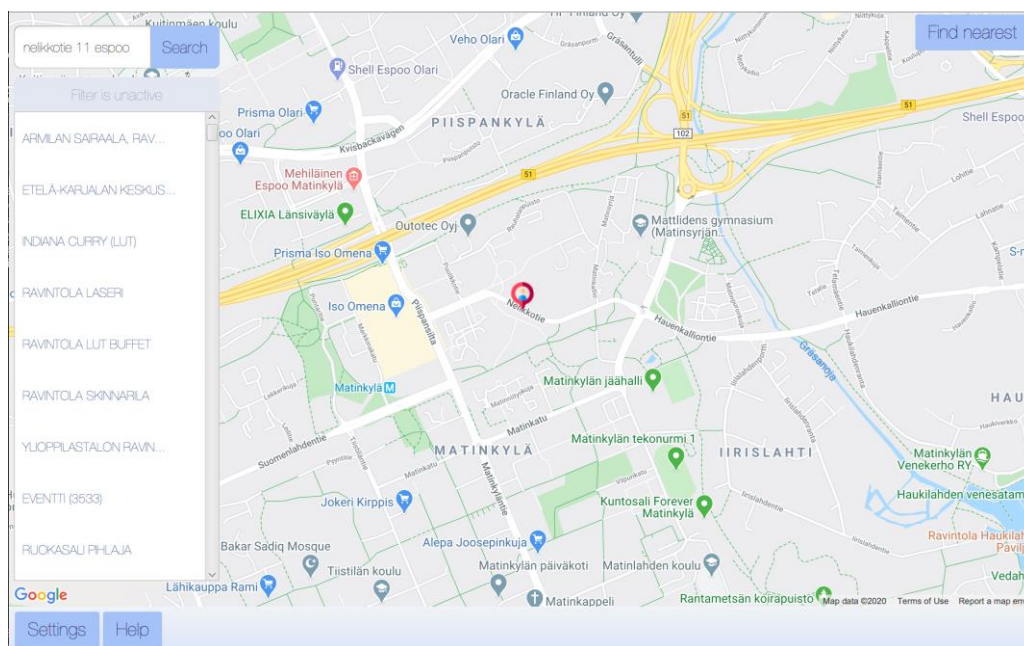
Kuva 9: Sovelluksen aloitusnäkö

### 5.2 Kartta

Ravintoloita voi sovelluksessa hakea kartalla liikkumalla. Karttaa voi lähentää ja loitontaa hiiren rullaustoiminnolla ja liikkuminen onnistuu klikkaamalla jotakin kohtaa kartalla ja hiirtä liikuttamalla.

### 5.3 Sijainnin etsiminen

Sovelluksella voi etsiä sijainnin kartalta. Hakusana on suotavaa kirjoittaa muodossa 'Katuosoite 1, Kaupunki', jotta oikea sijainti löytyy varmemmin. Osoitteen kirjoittamisen jälkeen Search-painiketta painamalla kartta keskittyy kyseiseen sijaintiin ja kartalla asettuu ikoni kyseiseen sijaintiin.

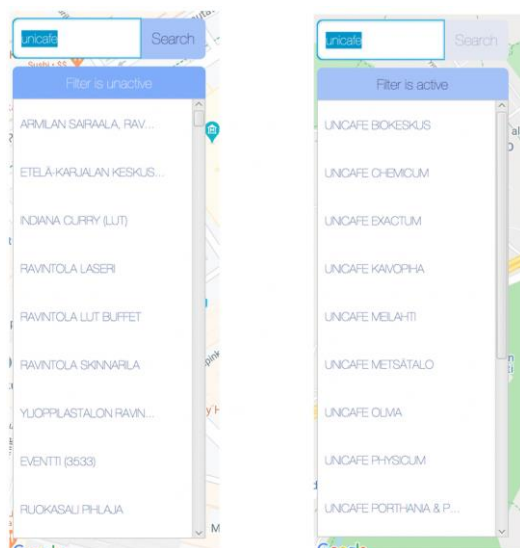


Kuva 10: Sijainnin etsiminen

### 5.4 Ravintolalistan suodattaminen & ravintolan lisätiedot

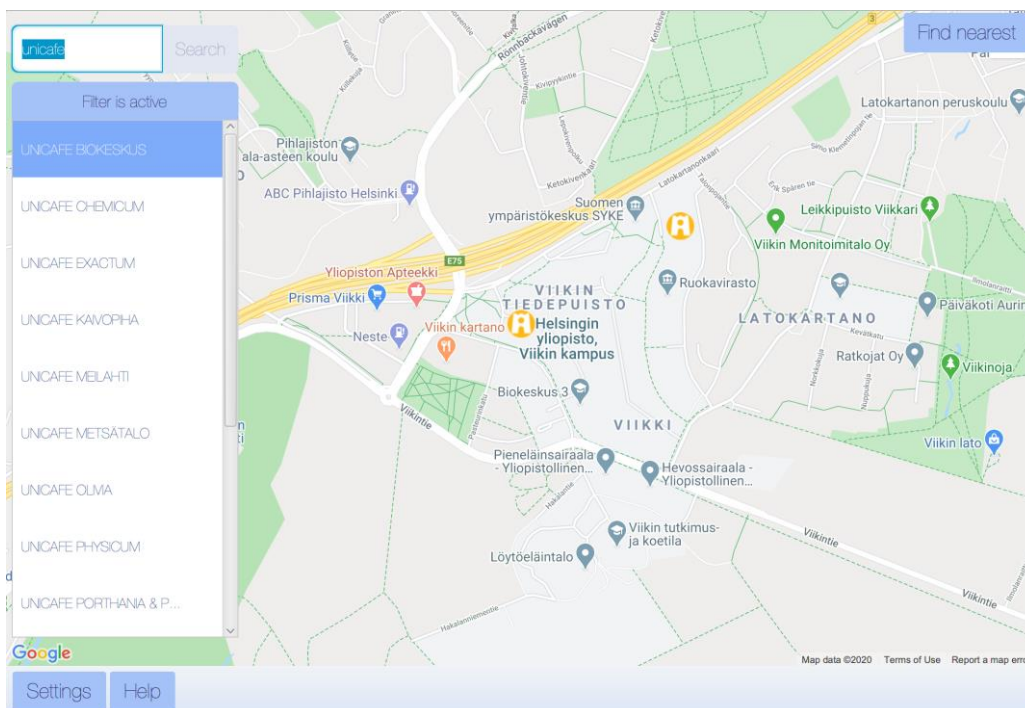
Hakukentän muuttaminen ravintolahakuun sopivaksi tapahtuu painamalla Filter is unactive-painiketta, jolloin sen tekstiksi muuttuu Filter is active. Suodatustilassa hakukenttään kirjoitettu hakusana muokkaa ravintolalistaa joka napautuksen perusteella suodattaen listalta pois tulokset, jotka eivät täsmää hakusanaan. Haku kohdistuu ravintoloiden nimeen, osoitteeseen ja paikkakuntaan.





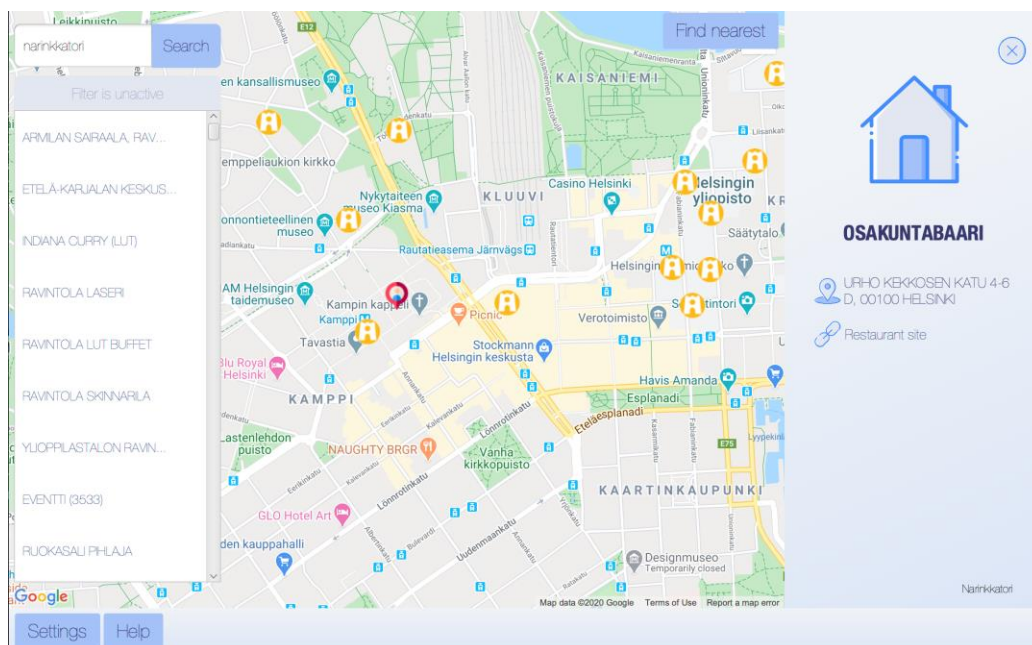
Kuva 11: Ravintolalistan suodatus hakusanalla

Mikäli hakutuloksia on alle 20, keskittyy kartta näyttämään listan ensimmäisen ravintolan kartan keskipisteenä. Yli 20 hakutuloksella kartta ei muutu. Listalta voi valita haluamansa ravintolan, jolloin sen ravintolan sijainti näytetään kartalla.



Kuva 12: Ravintolalistan suodatus – hakutuloksia alle 20

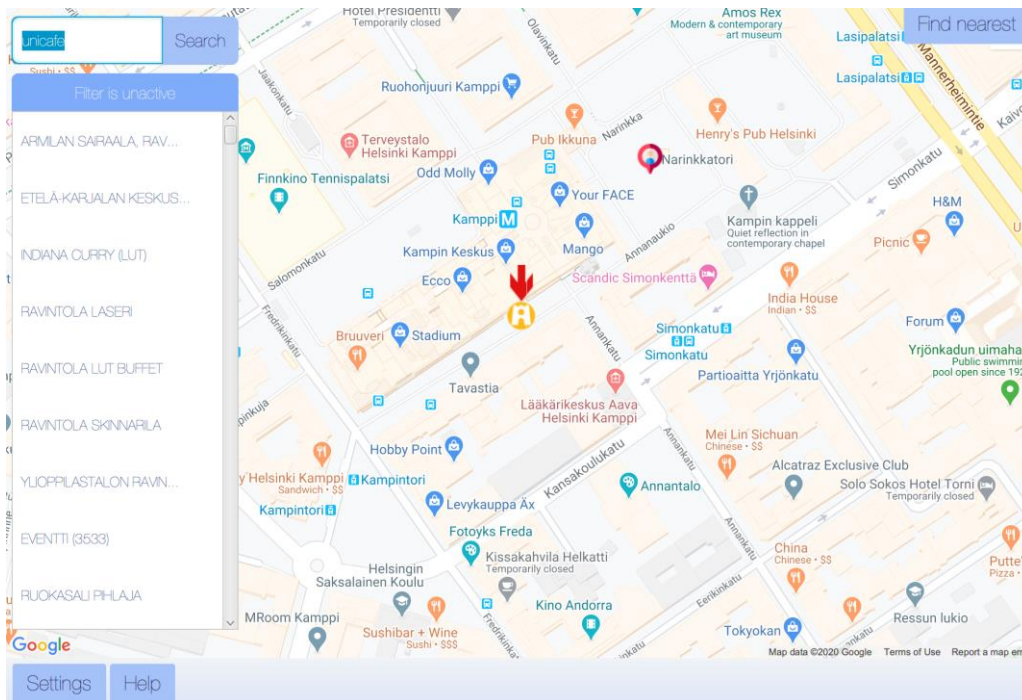
Kun listasta valitsee ravintolan osoittamalla sitä, avautuvat ravintolan lisätiedot oikean reunan sivupalkkiin. Mikäli ravintolan tiedoista löytyy ravintolan web-sivuston osoite, on se avattavissa sivupalkin Restaurant site -linkistä. Linkki avautuu käyttämäsi päätelaitteen oletusselaimeen. Mikäli käyttäjän sijainti on aiemmin määritelty, näkyy se sivupalkin alareunassa.



Kuva 13: Valitun ravintolan lisätiedot

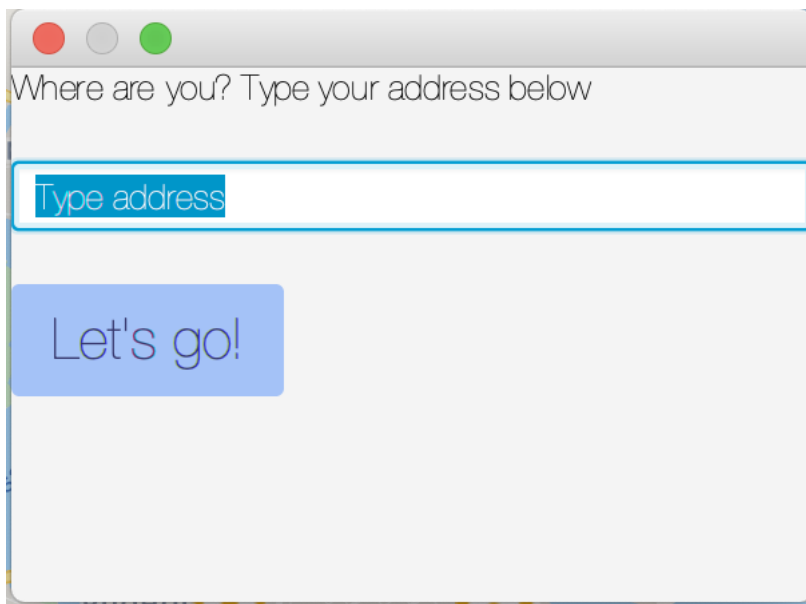
## 5.5 Lähimmän ravintolan etsiminen

Lähimmän ravintolan etsimisen edellytyksenä on, että sijainti, josta käsin lähintä ravintolaa etsitään, on tiedossa. Lähtösijainnin määrittely tapahtuu osoitehaun kautta ja tämä ilmenee punareunaisella kuvakkeella kartalla ja viimeisin haku tallennetaan automaattisesti. Sovelluksen oikeassa yläreunassa on painike lähimmän ravintolan etsimiseksi, jota painamalla sovellus laskee lähimmän ravintolan ja näyttää sen kartalla. Lähintä ravintolaa merkitään kartalla nuolikuvakkeella.



Kuva 14: Lähin ravintola kartalla osoitetaan punaisella nuolella

Mikäli osoitehakua ei ole suoritettu ennen Find nearest -painikkeen painamista, avautuu ponnahdusikkuna, jossa lähtösijainti pyydetään täyttämään. Let's go -painiketta painamalla lähin ravintola näkyy kartalla kuten kuvassa 14.

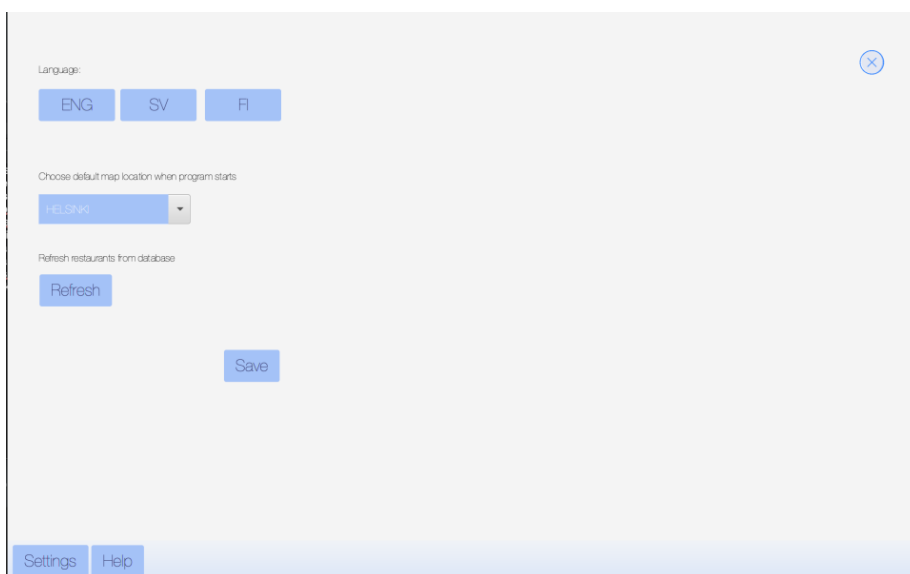


Kuva 15: Lähtösijainnin määrittäminen

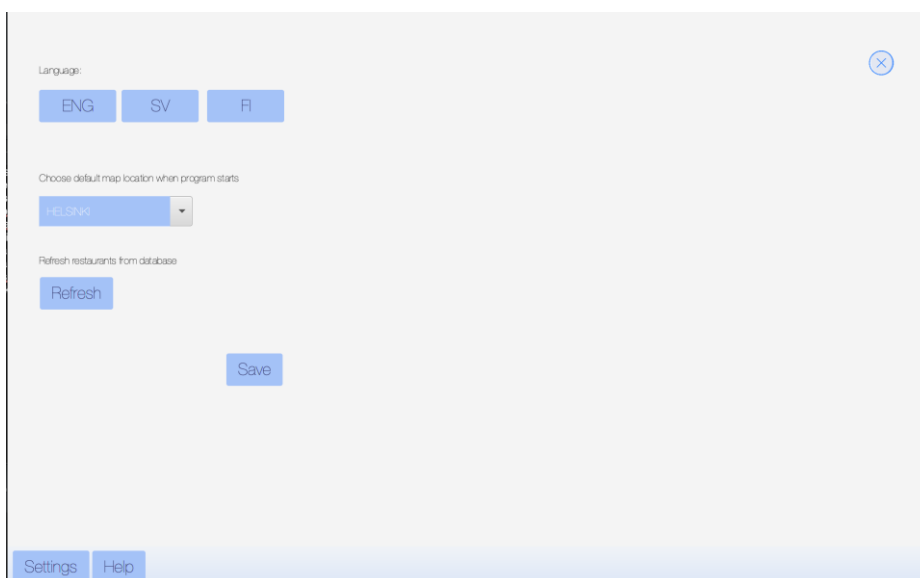
## 5.6 Omien asetusten määrittäminen

Asetuksissa on mahdollista määritellä kielivalinta ja oletuskaupunki. Kieli valitaan painamalla haluttua kielipainiketta. Oletuskaupunkimäärittelyllä voi vaikuttaa siihen, mikä kaupunki näkyy sovelluksen avautuessa oletusnäkyminä. Oletuskaupungin valinta tapahtuu pudotusvalikosta valitsemalla haluamansa kaupungin. Kaupungit listalla ovat sellaisia, joista löytyy Kelan ateriatukema ravintola.

Muutokset astuvat voimaan ja tallentuvat pysyvästi painamalla Save-painiketta. Asetusnäköymästä poistutaan painamalla oikean yläkulman rasti- tai Settings-painiketta.



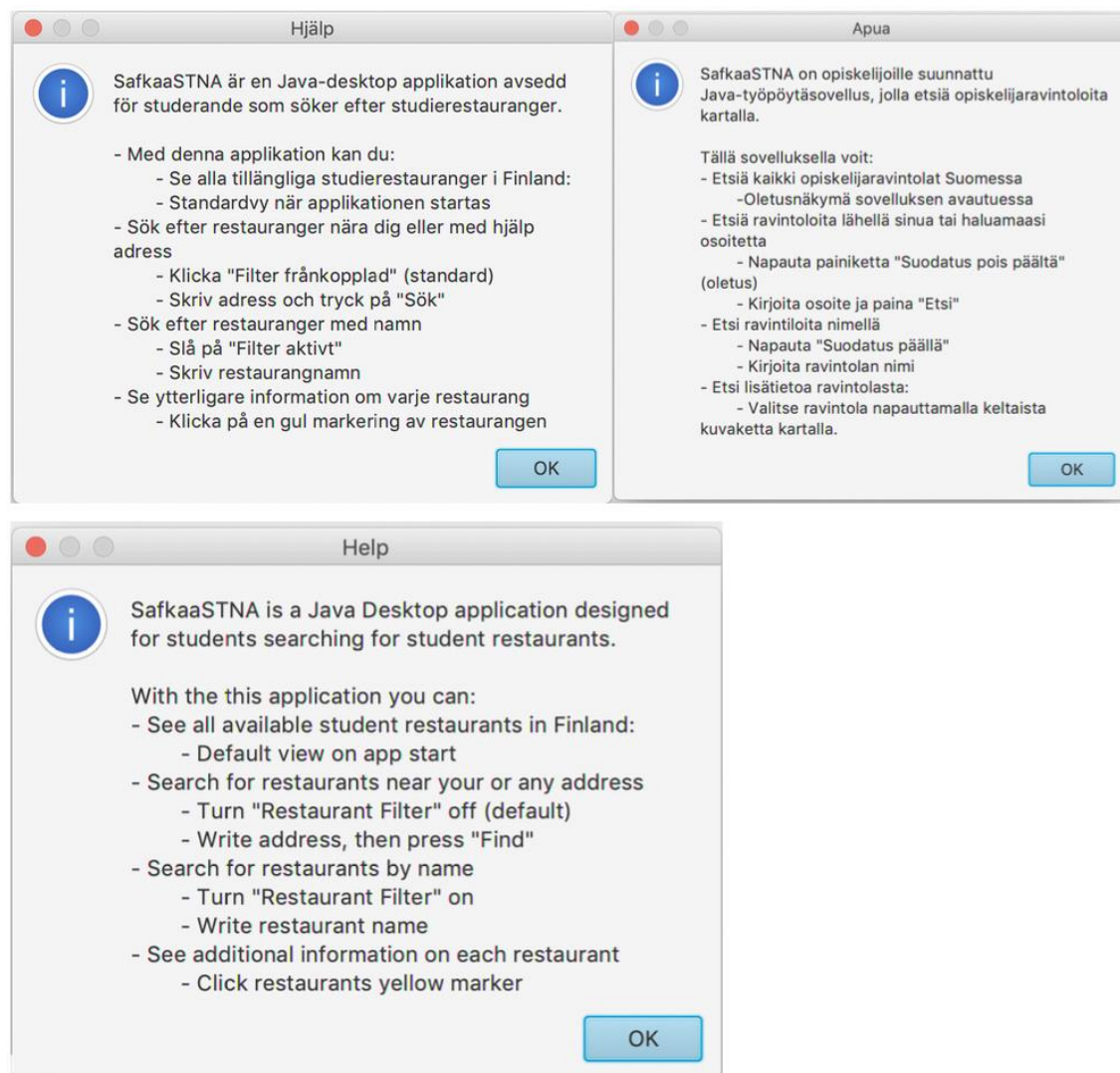
Kuva 16: Sovelluksen asetukset englanniksi



Kuva 17: Sovelluksen asetukset ruotsiksi

## 5.7 Ohjeet

Sovelluksen suppea käyttöohje on avattavissa milloin tahansa alapalkin Help-painikkeesta.



Kuva 18: Ohjetoiminto kaikilla kielivaihtoehdoilla

## Lähteet

Opiskelijaravintolahaku – kela.fi. 2016. Verkkoaineisto. < <https://www.kela.fi/ateriatuki-opiskelijaravintolahaku>>. 15.6.2016. Luettu 23.1.2020.

Kelan tukemat opiskelijalounasravintolat kartalla - Ateriatuki.com. Verkkoaineisto. <<https://ateriatuki.com>>. Luettu 23.1.2020

mapInitialized() gets never called during high load on parallel thread · Issue #81 · rterp/GMapsFX. 2016. Verkkoaineisto. <<https://github.com/rterp/GMapsFX/issues/81>> 04.07.2019. Luettu 21.2.2020.