



Jimi Koivuoja

# Paikkatietojärjestelmän toteutus CRM-järjestelmän tueksi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Maanmittaustekniikka

Insinöörityö

10.5.2022

## Tiivistelmä

Tekijä: Jimi Koivuoja  
Otsikko: Paikkatietojärjestelmän toteutus CRM-järjestelmän tueksi  
Sivumäärä: 41 sivua + 3 liitettä  
Aika: 10.5.2022

Tutkinto: insinööri (AMK)  
Tutkinto-ohjelma: maanmittaustekniikka  
Ohjaajat: lehtori Jussi Laari

---

Toimiva projektien- ja asiakkuuksien hallintajärjestelmä (CRM-järjestelmä) on tehokas työkalun liiketoiminnan edistämiseen. CRM-järjestelmällä ei ainoastaan ylläpidetä asiakasprofilleja sekä projektien tietoja, vaan järjestelmä myös laskee keskeisiä suorituskykyindikaattoreita, joiden avulla yrityksen johto voi seurata toiminnan kehitystä. Oikein käytettynä CRM-järjestelmä on jo yksinään hyödyllinen, mutta kehitysprojektin muodossa lähdettiin tutkimaan, että voisiko CRM-järjestelmän käyttöä edistää paikkatiedon avulla.

Yrityksessä oli aiemmin pilotoitu paikkatiedon yhdistämistä projektitietojen kanssa, mutta pilottiratkaisussa tietojen ylläpito osoittautui työlääksi ja karttatiedot eivät pysyneet ajantasaisina. Automatisoidun ja helppokäyttöisen ratkaisun löytämiseksi ryhdyttiin tutkimaan paikkatietointegraation toteutusmahdollisuuksia. Markkinoilla ei ollut saatavilla valmista integraatiota, joka olisi ollut yhteensopiva CRM-järjestelmän kanssa. Kehitysprojektia laajennettiin ja paikkatietojärjestelmää lähdettiin toteuttamaan yrityksessä sisäisenä sovelluskehityksenä.

Aluksi tutkimusvaiheessa osoitettiin paikkatietojärjestelmän toteutustapa kelvolliseksi. Paikkatietojärjestelmän rakentuu ulkoisen palveluntarjoajan CRM-järjestelmän tietokannasta, yrityksen hallinnoimasta MongoDB-paikkatietokannasta sekä paikkatietosovelluksesta, joka liittää CRM-järjestelmän kirjauksille paikkatiedon käyttäjän syöttämän lähiosoitteen perusteella. Paikkatietojärjestelmä vaihtaa tietoa CRM-järjestelmän kanssa sen REST-ohjelmointirajapinnan kautta ja luo CRM-kirjauksia vastaavat dokumentit paikkatietokantaan. Dokumenteille tallennetaan sijaintitiedot ja muut avaintiedot esitettävässä muodossa. Paikkatietosovellukseen toteutettiin myös lisätoiminnallisuuksia, kuten sijainnin syöttömoduuli, jolla käyttäjä voi vaihtoehtoisesti tulkita projektille sijainnin siirtämällä kartalla merkin hankkeen sijaintiin.

Paikkatietojärjestelmän ensimmäisen kokoonpanon toiminnallisuus vahvistettiin tietokantojen tarkastusoperaatioilla ja käyttäjäkokeilulla, jossa paikkatietojärjestelmä osoittautui menestykseksi. Järjestelmä on seuraavaksi määrä ottaa käyttöön, jonka jälkeen projekti siirtyy jatkokehitys- ja ylläpitovaiheeseen.

Avainsanat: paikkatietojärjestelmä, asiakkaiden- ja projektienhallintajärjestelmä, REST-ohjelmointirajapinta, paikkatietokanta, JavaScript, verkkosovellus

## Abstract

Author: Jimi Koivuojja  
Title: Creating GIS to Complement CRM System  
Number of Pages: 41 pages + 3 appendices  
Date: 10 May 2022

Degree: Bachelor of Engineering  
Degree Programme: Land Surveying  
Supervisors: Jussi Laari, Principal Lecturer

---

The final year project aimed at creating a software tool that would combine location data with the information of a Customer Relationship Management (CRM) system to allow the projects of the commissioning company to be seen on a map. To achieve this, the CRM and potential technologies for implementation were studied and code written.

The spatial data cue is put into the CRM system data record as an address, and the cue is then converted into coordinates. The spatial data and the most essential project information from the CRM system entries were stored in a separate non-relational MongoDB spatial database as documents. Data modifications between the systems were handled with HTTP requests, routed through each system's own application programming interfaces.

The goals set for the project were achieved, and the GIS now functions as intended. The GIS bolsters the CRM system with intuitive spatial data workflows, which notably reduce manual data validation. The GIS is now ready to be deployed company-wide, and any business utilizing the same CRM system can integrate the GIS to work alongside their CRM environment as well.

Keywords: GIS, CRM, REST API, geographic information database, JavaScript, web application

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Sovellus	4
2.1	Ohjelmointikieli	6
2.2	Suoritusympäristö	7
2.3	Työpöytäsovellus	7
3	Tietokannat	9
3.1	Tietokanta	9
3.2	Paikkatietokanta	10
3.3	Tietokantojen yhdistäminen	13
4	Sovelluksen osiot	15
4.1	Käynnistys	15
4.2	Kartta	19
4.2.1	Karttanäkymä	19
4.2.2	Tietoikkuna	22
4.3	Tunnisteet	25
4.4	Sijainnin syöttö	28
5	Käyttäjäkokeilu	30
6	Seuraavat askeleet	32
6.1	Pohjatutkimustietomoduuli	32
6.2	Karttatoiminnot	34
6.3	Verkkokartta	35
7	Yhteenveto	37
	Lähteet	39

## Liitteet

Liite 1: Paikkatietojärjestelmän käyttöönottosuunnitelma

Liite 2: Tietokantatoimintojen tarkastusluettelo

### Liite 3: Paikkatietokannan dokumentin muotoilu

## Lyhenteet

- CRM: *Customer Relationship Management*. Järjestelmä, jolla hallinnoidaan asiakaskontakteja ja projekteja myyntityön edistämiseksi.
- CSS: *Cascading Style Sheets*. Tyylitiedoston koodiasu, joka noudattaa avoimen verkon W3C-määrittelyksiä. Koodiasulla määritellään, miltä HTML-koodatut kohteet näyttävät esimerkiksi verkkosivuilla.
- EPSG: *European Petroleum Survey Group*. Tieteellinen organisaatio, joka dokumentoi tarkasti koordinaatistoihin liittyviä tietoja ja julkaisi niitä. Organisaation nimeä kantavat EPSG-koodit, joilla tunnistetaan koordinaatistoja toisistaan. Alkaen vuodesta 2005 EPSG-koodien tietokantaa on ylläpitänyt International Association of Oil and Gas Producers Surveying and Position Committee (OGP).
- HTML: *Hyper-Text Markup Language*. Vakioitu merkintäkieli, jolla määritellään verkkosivun rakenne ja sisältö.
- HTTP: *Hypertext Transfer Protocol*. Verkkosivujen ja -palvelimien väliseen tiedonvaihtoon kehitetty tiedonvaihdon yhteyskäytäntö.
- JSON: *JavaScript Object Notation*. Tekstimuotoinen ja selkokielineen tiedonvaihtoformaatti, joka perustuu JavaScriptin vakiointimäärittelyihin.
- KML: *Keyhole Markup Language*. XML-vakiointia noudattava merkintäkieli, jolla maantieteellistä tietoa voidaan esittää esimerkiksi Google Earthissa.
- KMZ: *Keyhole Markup Language Zipped*. Tiedostoformaatti, joka on paketoitu versio maantieteellistä tietoa sisältävästä KML-tiedostosta. Paketoimalla tiedoston koko saadaan pienennettyä kymmenykseen.

- KPI: *Key Performance Indicator*. Keskeinen suorituskykyindikaattori on arvo, jolla terävöitetään tavoitteita ja tuetaan päätöksen tekoa.
- NoSQL: *Not Only SQL*. Tietokanta, joka ei ole ainoastaan relaatiotietokanta.
- REST: *Representational State Transfer*. Muotoilusääntö ohjelmointirajapinnan pyyntöjen ja vastausten vaihtoon, jonka mukaan pyynnöt toimitetaan usein JSON-muodossa. Ohjelmointirajapinta voidaan todeta REST-periaatteita vastaavaksi (RESTful).
- SOAP: *Simple Object Access Protocol*. Ohjelmointirajapinnan pyyntöjen tiedonvaihdossa käytettävä XML-rakenteeseen perustuva sekä vakioitu muotoilusääntö
- SQL: *Structured Query Language*. Tietokantojen tiedonhakuun ja -manipulointiin tarkoitettu kieli.
- URL: *Uniform Resource Locator*. Verkko-osoite, joka yhdistää käyttäjän verkossa julkaistuun resurssiin.
- UUID: *Universal Unique Identifier*. Yksittäiselle kohteelle osoitettava universaali ja ainutlaatuinen merkkijono, jolla kohde voidaan tunnistaa saman kaltaisten kohteiden joukosta.
- XML: *Extensible Markup Language*. Merkintäkieli, jossa käyttäjä voi itse räätälöidä tunnisteet palvelemaan kohdetarkoitusta. XML-muotoiltu tiedosto voidaan jäsenellä sopimaan mihin tahansa käyttöympäristöön, sillä XML-syntaksi on vakioitu.

## 1 Johdanto

Laaja projektikanta on yksi tärkeimmistä arvoista konsulttialan yrityksen elinvoimaisuudelle ja kannattavuudelle. Opinnäytetyön tilaajayrityksellä Sipti Infralla on jatkuvasti aktiivisena kymmenittäin erityyppisiä pääosin geotekniikan ja infra-alan projekteja pitkin Suomea painottuen maan eteläisiin osiin. Projektien toteutusaikavälit vaihtelevat muutamista viikoista vuosiin, ja projektikanta kasvaa vuositasolla keskimäärin 150:llä. Yrityksellä on käytössä asiakkaiden- ja projektienhallintaan ulkoisen palveluntarjoajan CRM-järjestelmä (engl. Customer Relationship Management), jossa projektit alkavat tarjousvaiheella ihanteellisesti päättyen tilauksiksi. Projekteille kirjataan kaikki myyntiin ja projektin ominaispiirteisiin liittyvät tiedot, sillä ne tukevat projektien kehitystä ja avustavat yrityksen toiminnan kannattavuuden seuranta.

Tarkemmin ottaen Sipti Infran projektit profiloituvat lähinnä uudis- ja korjausrakennushankkeisiin, mutta ovat silti moniulotteisia esimerkiksi kohderakennuksen ja suoritettavan työtehtävän osalta. Nykytilassaan projekteja ei voi CRM-järjestelmässä erottaa toisistaan muun kuin nimen ja numeron perusteella, jolloin projektien vertailu ominaisuustietojen perusteella on lähes mahdotonta. Projektien tietoja voitaisiin hyödyntää analysointitarkoituksissa paremmin, mikäli projektien tiedot täydennettäisiin huolellisesti ja syventäviä tietoja kirjattaisiin talteen mahdollisimman paljon. CRM-järjestelmässä tiedon määrä on valtava, ja projektien kohteet sijaitsevat maantieteellisesti eri paikoissa. Jokaiselle projektille olisikin houkuttelevaa osoittaa sijainti, kirjata se muistiin ja esittää projektit karttaesityksenä, jossa järjestelmän tietoja voitaisiin tuoda esille ja tarkastella kokonaisuutta maantieteellisessä asiayhteydessä. CRM-järjestelmä ei kuitenkaan mahdollista sijainnin tulkintaa, eikä järjestelmässä ole karttaliittymää.

Projektien karttamuotoinen esitystapa on aiemmin yrityksessä koettu tärkeänä tarkasteluperspektiivinä, ja siksi sitä on pilotoitu. Pilottiratkaisussa oli onnistuneesti osa projektitietokannasta julkaistu Googlen loppukäyttäjille suunnatussa MyMaps-selainsovelluksessa, jonne oli syötetty projektien sijaintien lisäksi avaintietoja sekä kuvauksia tehdyistä työtehtävistä.

Kartalle oli kohteita tuotu CRM-järjestelmästä käytännössä manuaalisesti syöttämällä tietoja uudelleen, jolloin niiden johdonmukaisuus oli riippunut aineisto- ja tulkinneen työntekijän harkinnasta. Satunnaiset kirjaamisvirheet esimerkiksi tunnistetietojen kirjoitusasuissa eivät olleet ennen kuulumattomia, sillä tiedot eivät noudattaneet tiettyä vakio- tai muotoilua. Lopputulos oli staattinen karttaesitys, jossa esitetyt kohteet eivät välttämättä olleet vertailukelpoisia keskenään. Pilottiratkaisun viimeiseksi kompastuskiviksi osoittautui kirjaamisen ja kartan ylläpidon resurssi-intensiteetti ja se, että kartan ensimmäisen toteutuksen jälkeen ei voitu olla varmoja tietojen ajantasaisuudesta tai oikeellisuudesta.

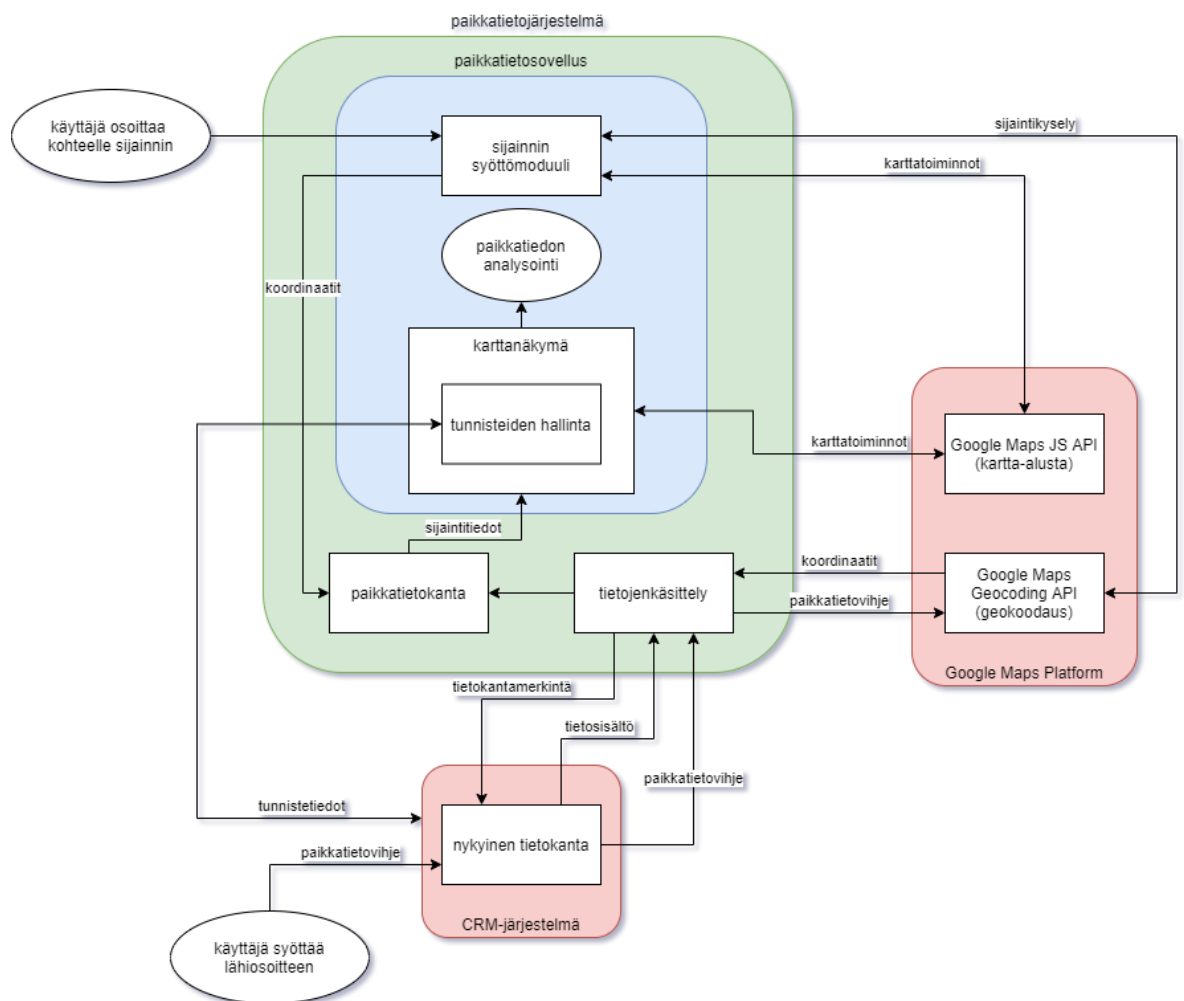
Pilottitoteutuksen ydinidea, eli projektien esittäminen kartalla on kuitenkin toteutuskelpoinen idea oikeilla työkaluilla, ja kohteiden paikkatiedosta olisi mahdollista saada irti muutakin kuin pelkkä hyvännäköinen karttaesitys. Käyttäjät voisivatkin esittämistoimintojen lisäksi tehdä paikkatietoanalyyskejä sekä rajauksia suoraan kartalla ja mahdollisesti luoda valintoihin perustuvia automaattisesti koottuja raportteja. Karttaesityksen luonnin ja sijainnin tulkinnan automatisoinnilla sekä tietojen poiminnalla suoraan CRM-järjestelmästä vältyttäisiin kaikilta mahdollisilta tiedonsiirrossa tapahtuvilta virheiltä ja automatisoinnilla minimoitaisiin myös kartan ylläpidon resursoinnin tarve. Idean jatkokehittämiseksi yrityksessä polkaistiin käyntiin kehitysprojekti insinööriyön muodossa, jonka tavoitteena oli luoda ajantasainen paikkatietokanta sekä sovellus karttaesityksen tarkasteluun. Lopputuotteena on paikkatietojärjestelmä, joka toteutetaan neljässä eri vaiheessa, joista tutkimus- ja toteutusvaiheet kuuluvat insinööriyön laajuuteen. Paikkatietojärjestelmän käyttöönotto sekä ylläpito- ja kehitysvaihe toteutetaan insinööriyön valmistumisen jälkeen. Kaikki vaiheet on havainnollistettu prosessikaaviona (kuva 1).



Kuva 1. Kehitysprojektitokonaisuus prosessikaaviona.

## 2 Sovellus

Kehitysprojektin pääasiallisena lopputuotteena on käyttäjälähtöinen ja web-pohjainen paikkatietosovellus, joka automatisoi sijaintitiedon tulkinna ja tallentaa sen oikealle CRM-järjestelmän kirjaukselle. Sovellukseen olisi mahdollista jatkossa liittää muitakin tulevia koodausprojekteja, sillä sovellus toteutetaan modulaarisena. Paikkatietojärjestelmän toimintakaavio on esitetty kuvassa 2.



Kuva 2. Paikkatietosovelluskokonaisuuden toimintakaavio.

Kehitysprojektin tutkimusvaiheessa markkinoilla ei ollut saatavilla yrityksen tarpeisiin ja nykyiseen CRM-järjestelmään integroitavaa paikkatietotuotekokonaisuutta. Siksi paikkatietojärjestelmä päätettiin toteuttaa yrityksen sisäisenä ohjelmistoprojektina, joka mahdollistaa myös sovelluksen suunnan ohjaamisen

ja kokonaisuuden räätälöinnin juuri yrityksen tarpeisiin sopivaksi. Paikkatietojärjestelmää voidaan sisäisen ohjelmistokehityksen ansiosta jatkokehittää joustavasti ja kehitys- sekä ylläpitokustannukset voidaan arvioida tarkemmin kehitysprojektin toteutukseen käytettyjen resurssien perusteella. Yrityksen sisäisessä sovellustuotekehityksessä on omat tunnistettavat riskinsä. Niihin lukeutuvat kehitysryhmän kokoonpanon muutokset, osaaminen ja sen pysyvyys, mikä voi johtaa osaamisen hankintaan muualta ja sitä kautta johtaa korkeisiin kustannuksiin (1).

Web-pohjainen toteutustapa valittiin kehitysryhmän osaamiskokonaisuuden lisäksi sille ominaisten etujen vuoksi. Web-aplikaatio voidaan esimerkiksi päivittää keskitetysti palvelimelta verkon ylitse, jolloin loppukäyttäjillä on aina paikkatietosovelluksen ajantasaisin kokoonpano käytössä. Päivitysten hallinnan lisäksi web-aplikaatio voidaan julkaista mobiilikäyttöympäristössä, jolloin sovellusta on mahdollista käyttää missä tahansa, milloin tahansa. Mobiilikäyttöä ei toistaiseksi suunniteltu, mutta mahdollisuudet antavat hyvän kuvan web-aplikaatioiden joustavuudesta ja skaalautuvuudesta. (2)

Web-aplikaatiot ovat yleisesti riippuvaisia internetyhteydestä, kun sovellus on saavutettavissa verkossa ja tietoja siirretään HTTP-protokollan ylitse. Internetissä web-aplikaatiot ovat haavoittuvaisia esimerkiksi Cross-site scripting (XSS) tai Cross-site request forgery (CSRF), joilla arkaluontoisiin tietoihin yritetään päästä suoraan käsiksi tai välillisesti applikaation käyttäjien avulla (3). Paikkatietojärjestelmässä on varmistettu, ettei arkaluontoisia tietoja voi päätyä ulkopuolisten käsiin. Paikkatietojärjestelmän toiminta on rajattu siten, että sensitiivisiä tietoja käsitellään ainoastaan suojattujen yhteyksien ylitse. Esimerkiksi tietoja noudettaessa CRM-järjestelmästä pyynnöissä käytetään kertakäyttöisiä salausavaimia, joiden tuottaminen vaatii käyttäjätunnuksen (engl. *Client ID*) ja salatun avaimen (engl. *Client Secret*). Fyysinen tietoturvariski on olemassa, sillä ulkopuolinen henkilö voi päästä fyysisesti todellisen käyttäjän työpisteelle ja sitä kautta järjestelmään. Riski on pieni, ja sen rajoittamiseksi järjestelmään olisi mahdollista luoda käyttäjienhallinta, jonka avulla oikeuksia järjestelmään voitaisiin jakaa ja käyttäjät voitaisiin tunnistaa. Käyttäjienhallinta on tarkoitus toteuttaa

sovelluksen jatkokehitysvaiheessa, mutta se ei ole prioriteettilistalla kovin korkealla, sillä tietoturvariski on nykymuodossaan hallittavissa.

## 2.1 Ohjelmointikieli

Paikkatietosovelluksen käyttäjälle näkyvä osuus (engl. *Front-End*) koostuu periaatteessa verkkosivuista, jolloin niiden määrittelyssä on käytetty HTML-merkintäkieltä (engl. *Hyper-Text Markup Language*), jolla kuvataan verkkosivun rakennetta sekä välitetään rakenteen muotoilun tiedot selaimelle. HTML-elementtien tyyllittely tehtiin erillisiin CSS-tyylitiedostoihin (engl. *Cascading Style Sheets*) (4; 5). Sovelluksen näkymien toiminnallisuudet toteutettiin JavaScriptilla, jossa hyödynnettiin siihen liitettyjä lisäkirjastoja. JavaScript on järjestyksessä toimiva (engl. *synchronously*) korkean tason objektorientoitunut ohjelmointikieli, jonka muotosääntelyä, eli syntaksia on helppo ymmärtää selkokielistikin (6; 7).

Ohjelmointikielen valintaan vaikutti ennen kaikkea JavaScriptin yleisyys, sillä se on Google Trendsiin perustuvan ohjelmointikielten yleisyysindeksin (engl. *PYPL-index*) perusteella maailman kolmanneksi suosituin ohjelmointikieli (4). Sen lisäksi JavaScriptin ymmärrettävyys on eduksi koko projektia tarkasteltaessa. Esimerkiksi kehitystiimin kokoonpanon muuttuessa seuraavat koodia käsittelevät henkilöt todennäköisesti voivat jatkaa tuotekehitystä, sillä JavaScript jakaa Java- ja C++ -ohjelmointikielten kanssa samankaltaisen syntaksin (5).

JavaScriptin rinnalla paikkatietosovelluksen koodissa on käytetty lisäkirjastoina JQueryä sekä Lodashia, ja niillä kummallakin on oma tärkeä roolinsa sovelluksen toiminnallisuuksissa. Ohjelmointirajapintojen välisessä tiedonvaihdossa käytettiin pääasiallisesti JQueryä, mutta sitä hyödynnettiin myös sovelluksen käyttäjälle näkyvän osan toteutuksessa. Lodashia vuorostaan käytettiin tietojenkäsittelyfunktioissa, koska sillä pystyttiin korvaamaan pitkiäkin tavanomaisella JavaScriptillä kirjoitettuja koodiosoita siisteiksi muutaman rivin osioiksi (esimerkikoodi 1). Lodashilla onnistuneesti selkeytettiin koodia helppolukuisemmaksi ja

Lodashin taulukkovertailu- sekä muotoilumetodit osoittautuivat erityisen hyödyllisiksi.

```
let forManualInput = _.filter(parsedData, item =>
  _.isEmpty(item.description) ||
  _.isNull(item.description)
)
```

Esimerkkikoodi 1. Lodash-kirjastosta valikoituja metodeja käytettiin paikkatietosovelluksen tietojenkäsittelyfunktioissa. Esimerkin funktiossa taulukko-objektista valitaan kaikki objektit, joiden kuvauksen arvo on joko tyhjä tai nolla. Tiedot tallentuvat uuteen *forManualInput*-muuttujaan.

## 2.2 Suoritusympäristö

Paikkatietosovelluksen ytimessä toimii tehokas ja skaalautuva Node.js -suoritusympäristöä (jäljempänä Node), joka soveltuu web-aplikaatioiden toteutukseen. Node käyttää nimensä js-päätteen mukaisesti ohjelmointikielenä JavaScriptia, ja sen ansiosta sovellukseen kirjoitettua koodia voidaan käyttää uudelleen esimerkiksi kartan verkkojulkaisussa. Noden rakenne hyödyntää tehokasta V8 JavaScript -moottoria ja toimii tavanomaisen selainympäristön tavoin yksisäikeisenä prosessina. Nodessa on kuitenkin hyödynnetty periaatteita, kuten vastakutsufunktioita (engl. *Event-Handlers*) ja tapahtumasilmukkaa (engl. *Event-Loop*), joiden avulla koodia on mahdollista suorittaa tahdittomasti (engl. *synchronously*). Nodella voidaan käsitellä tuhansia palvelinyhteyksiä ja pyyntöjä ilman, että ne kuormittavat käyttöympäristöä liikaa rokottamalla prosessitehoja. (10; 11.)

## 2.3 Työpöytäsovellus

Jo kehitysprojektin tutkimusvaiheessa loppukäyttäjiltä saatiin palautetta, että paikkatietosovelluksen tulee olla työpöytäsovellus, sillä se koettiin luotettavampana ja tuttuna kokonaisuutena. Sovelluksen koodissa käytettävä JavaScript ei varsinaisesti ole työpöytäsovelluksille suunnattu ohjelmointikieli, mutta Noden ja siihen liitetävän Electron.js -kehysympäristön (jäljempänä Electron) avulla web-aplikaation paketointi työpöytäsovellukseksi on mahdollista. Electron on

käytännössä selainympäristö, sillä sen sovellusarkkitehtuuri perustuu avoimen lähdekoodin selainprojektiin: Chromiumiin (6). Paikkatietosovelluksen varsinainen julkaisu tehdään käyttöönottovaiheessa käyttöönottosuunnitelman (liite 1) mukaisesti.

Sovellus olisi voitu toteuttaa web-aplikaationa, jonka resurssit olisivat saavutettavissa tietyssä URL-osoitteessa. Tosin silloin käyttäjät saattaisivat yrittää käyttää sovellusta selaimella, joka on vanhentunut tai ei tue sovelluksen kaikkia toiminnallisuuksia täysimääräisesti. Eri selainten välisiltä yhteensopivuusongelmilta vältytään, eikä niitä tarvitse huomioida, kun sovellus lukitaan Electron-kehysympäristöön.

Käyttöönottoympäristössä laitteiden käyttöjärjestelmä on poikkeuksessa Windows, joten käyttöjärjestelmien välisiä yhteensopivuuksia ei ollut tarvetta huomioida Electronissa. Sovellusta ei myöskään olla toistaiseksi julkaisemassa muille käyttöjärjestelmille, mutta Electron kuitenkin jättää avoimeksi mahdollisuuden laajentaa julkaisua Mac ja Linux-käyttöjärjestelmille. Web-aplikaatio voidaan julkaista myös Android-mobiilikäyttöjärjestelmälle pienellä vaivalla esimerkiksi WebViewin avulla (7).

Electron perustuu Chromiumiin, joten paikkatietosovelluksen koodi lähtökohtaisesti suoritetaan järjestyksessä (engl. *Synchronously*). Paikkatietosovelluksen toteutuksessa on kuitenkin tavoiteltu mahdollisimman tahditonta (engl. *Asynchronous*) toimintaa JavaScriptin lupausten (engl. *Promises*) avulla. Niillä voidaan koodiosuoksien tuloksia jättää ratkaisematta tekemällä ns. lupaus funktiolle tuloksen palauttamisesta myöhemmässä vaiheessa, jolloin koodin ei tarvitse pysähtyä odottamaan funktion tuloksen muodostumista. (8)

## 3 Tietokannat

### 3.1 Tietokanta

CRM-järjestelmää pidetään tiedon oletustallennussijaintina, sillä sinne kirjataan mm. projektien tietoja sekä kaikki asiakkaiden tiedot asiakasprofiileille. CRM-järjestelmä tulkitsee käyttäjän syöttämiä tietoja ja luo niiden perusteella uusia tietotuotteita sekä luo automaattisia tietoja, kuten aikaleimoja sekä projektia yksilöiviä tunnisteita. Järjestelmän taustalla toimii tietokanta, josta tietoja voidaan hakea tarkasteltaviksi ja käsiteltäviksi avoimen ohjelmointirajapinnan kautta HTTP-pyyntöjen avulla. Paikkatietokannan toimivuutta tarkasteltiin ja sille suoritettiin testejä tarkastusluettelon avulla, johon on kirjattu kaikki toiminnallisuudet ja niiden odotetut tulokset (liite 2). Tarkastusluettelo voidaan hyödyntää myös sovelluksen käyttöönotossa, jolloin se voidaan käydä uudemman kerran läpi kaikkien tietokantatoiminnallisuuksien vahvistamiseksi.

Kehitysprojektin tutkimusvaiheessa CRM-järjestelmän palveluntarjoaja oli päivittävässä ohjelmointirajapintansa määrittelyä SOAPista (engl. *Simple Object Access Protocol*) vastaamaan nykyaikaisemman REST-ohjelmointirajapinnan (engl. *Representational State Transfer*) arkkitehtuurillisia määrittelyjä. Kehitettävän paikkatietojärjestelmän kannalta oleellisin ero SOAPin ja RESTin välillä on niiden pyyntöjen muotoilu. SOAP-pyyntöjen rakenne on XML-merkintäkielen muodossa, ja REST-pyyntöt voidaan muotoilla moniin eri tiedonvaihtoformaatteihin, joista yleisin ja paikkatietojärjestelmässäkin käytetty muoto on JSON (9). CRM-järjestelmän rajapinnan muutostyön aikataulu oli aiemmin pettänyt monien kertaan, ja kommunikaatio edistymisestä oli heikkoa. Aikataulun epävarmuus aiheutti vaikeuksia paikkatietojärjestelmän kehityksessä, sillä ei ollut varmaa tietoa siitä, milloin ohjelmointirajapinnan uudistukset tulevat voimaan. HTTP-pyyntöjä ei myöskään ollut mahdollista kirjoittaa koodiin valmiiksi vastaamaan REST-muotoilua.

Paikkatietojärjestelmää ei olisi ollut kannattavaa toteuttaa käyttämään SOAP-määrittelysten mukaisia pyyntöjä, joten järjestelmän kehitystä päätettiin pitää

pidossa hetken aikaa paremman tiedon puutteessa. Kehitysprojektin onneksi toteutusvaiheessa CRM-järjestelmän palveluntarjoajan rajapintamääritysten muutostyö oli sen verran pitkällä, että pääsimme liittymään REST-ohjelmointirajapinnan sovelluskehitysryhmään. Paikkatietojärjestelmän koodissa pyynnöt pystyttiin kirjoittamaan vastaamaan tulevia REST-määrittelyjä ja pyynnöt voitiin kohdistaa CRM-järjestelmän käyttökokeiluympäristön ohjelmointirajapintaan. Paikkatietojärjestelmän käyttöönottovaiheessa pyynnöt tulee käydä systemaattisesti läpi ja osoittaa ne todelliseen järjestelmäympäristöön ja sen tietokantapäätteisiin.

### 3.2 Paikkatietokanta

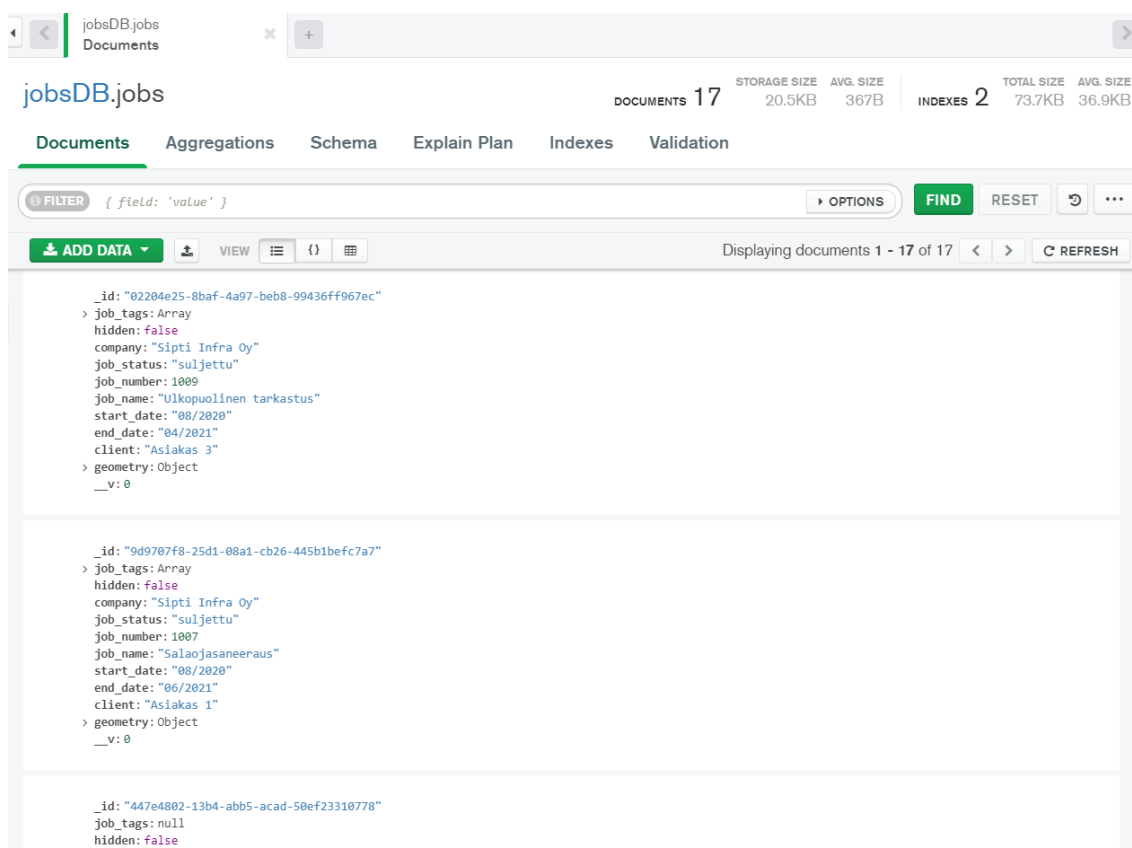
Projektien sijaintitiedot on tallennettava koordinaatteina johonkin järkevään ja saavutettavaan paikkaan. Koordinaatteja ei lähtökohtaisesti kannata syöttää CRM-järjestelmään, sillä silloin kirjausten esittäminen kartalla olisi riippuvainen CRM-järjestelmän yhteydestä. CRM-järjestelmän tietokannassa ei myöskään ole sijaintitiedoille valmista tallennuspaikkaa tai nimikettä, ja tietokannan rakenne noudattelee ulkoisen palveluntarjoajan määrittelemää skeemaa, eli vakiomuotoilua, johon järjestelmän käyttäjien on vaikea vaikuttaa. Lisäksi tietokantapyyntöjen määrää vähennetään merkittävästi tallentamalla sijaintitieto helpommin saavutettavaan paikkaan. Vähentämällä tietokantaan kohdistuvien HTTP-pyyntöjen määrää varmistutaan, että tietokannassa olevat tiedot ovat aina alkuperäisiä eivätkä ne ole muuttuneet esimerkiksi virhetilanteen seurauksena.

CRM-järjestelmän tietokannan rajoitteita kierrettiin luomalla oma paikallinen tietokanta, jonka skeemaa on mahdollista mukauttaa paikkatiedon tarpeita varten. Valitussa ratkaisussa CRM-järjestelmän tietokanta ei muutu paikkatietokannaksi, vaan uusi paikallinen tietokanta toimii paikkatietokantana ja tietokantojen tietosisällöt yhdistyvät toisiinsa paikkatietosovelluksessa. Kokonaisuutena syntyy paikkatietojärjestelmä, joka koostuu kahdesta tietokannasta ja paikkatietosovelluksesta.

Paikkatietokannan kirjaukset perivät CRM-järjestelmän tietokannan vastaavilta kirjauksilta yksilöivät tunnisteet (UUID), joiden avulla tietokantojen kirjaukset ovat saumattomasti yhdistettävissä toisiinsa. Kirjausta yksilöivä tunniste ei voi muuttua CRM-järjestelmän tietokannassa, vaan se voidaan ainoastaan kokonaan poistaa ja siksi se on oiva tietokantoja yhdistävä arvo. Paikkatietokantaan rekisteröivät kirjausten tiedot on valmiiksi muokattu haluttuun esitysmuotoon, jotta sovelluksen ei tarvitse tehdä tietojen muotoilua reaaliaikaisesti.

Paikkatietokanta toteutettiin MongoDB-tietokantana (jäljempänä Mongo), jossa CRM-järjestelmästä kloonattuja tietoja ja niille liitettyä paikkatietoa säilytetään. Mongoon kirjaukset tallennetaan dokumentteina, joiden muotoilu noudattaa JSON-tiedonvaihtoformaatin (engl. *JavaScript Object Notation*) periaatteita. Mongo on myös NoSQL-tietokanta (engl. *Not Only SQL*), eli Mongo ei ole SQL-tietokantojen (engl. *Structured Query Language*) tavoin ainoastaan relaatiotietokanta (10). Mongossa dokumenttien muotoilun ei tarvitse noudattaa tiettyä vakiomuotoilua, ja dokumenttien arvot voivat poiketa keskenään. Kehitysprojektissa päätettiin, että paikkatietokannan dokumentit noudattelevat ennalta määritettyä skeemaa, joka on esitetty liitteessä 3. Vakiomuotoilun etuna on, että tarvittavat tiedot tulee kirjattua dokumenteille sovitussa laajuudessa sekä muodossa. Mongo mahdollistaa, että uusilla dokumenteilla voi olla enemmän tieto objekteja ja niitä ei tarvitse vanhoille dokumenteille täydentää ollenkaan. (11)

Kehitysprojektin toteutusvaiheessa Mongo-tietokanta oli asennettu paikalliselle tietokoneelle, jotta sitä pystyttiin parhaiten hyödyntämään kehitystyön tarpeisiin. Mongoa pystytään hallinnoimaan komentotulkin avulla, mutta sen sijaan paikkatietokantaa ja sen dokumentteja käsiteltiin MongoDB Compass -sovelluksen graafisella käyttöliittymällä (kuva 3). Paikkatietokantaa ei ole tarkoitus muokata käsin tai valvoa sen käynnissäoloa, vaan paikkatietojärjestelmän lopullisessa kokoonpanossa paikkatietokanta sijaitsee paikallisella palvelimella, ja se asetetaan sinne jatkuvasti aktiiviseksi prosessiksi. Näin varmistetaan, että paikkatietokanta on aina kytkettynä ja saavutettavissa käyttöympäristössä jokaiselta työpisteeltä.



Kuva 3. Paikkatietojärjestelmän toteutuksessa tallennettuja dokumentteja tarkasteltiin MongoDB Compass -sovelluksen graafisella käyttöliittymällä.

Paikkatietojärjestelmässä paikkatietokannalla on oma ohjelmointirajapinta, joka HTTP-pyyntöjen avulla käsittelee ja manipuloi paikkatietokannan dokumentteja. Pyyntöjä kohdistetaan tiedon sijainnin URL-muotoiseen paikkatietokannan päätepisteeseen. Esimerkiksi karttanäkymään haetaan paikkatietokannasta esitettäväksi kaikki kirjaukset, joille on osoitettu koordinaattitieto (esimerkkikoodi 2). Paikkatietokannan ohjelmointirajapinta tukee mm. lisäämisen, päivittämisen ja poistamisen pyyntöjä ja nämä on esitettyinä taulukossa 1.

```

router.get("api/jobs/location", (req, res) => {
  Job.find({"geometry.coordinates": {$ne: [0,0]}}).then((jobs) => {
    res.send(jobs)
  })
})

```

Esimerkkikoodi 2. Ohjelmointirajapinnan URL-päätepisteen koodissa on käytetty Express.js-kirjaston välitystä, jolla määritellään, kuinka URL-päätepisteen tulee vastata GET-pyyntöön. Päätepisteessä "api/jobs/location" esitetään kaikki

paikkatietokannan dokumentit, joiden geometriaobjektin koordinaattitiedot eivät ole 0,0, eli niille kirjauksille on osoitettu muut koordinaatit. Kirjaukset paljastetaan URL-osoitteessa JSON-objekteina ja päätepisteen kirjaukset voidaan esittää esimerkiksi paikkatietosovelluksen kartalla.

Taulukko 1. Listaus paikkatietokannan ohjelmointirajapinnan mahdollisista HTTP-pyyntöistä ja niiden käyttötarkoituksista. URL-osoitteet vaihtelevat eri pyyntöjen välillä.

Pyynnön tyyppi	Toiminta	Käyttötarkoitus
GET	hakee paikkatietokannasta kirjaukset, joille ei ole osoitettu koordinaatteja	sijainnittomat kirjaukset esitetään sijainninsyöttö -moduulissa, jossa käyttäjä voi antaa kohteelle sijainnin
GET	hakee paikkatietokannasta kirjaukset, joille on määritetty koordinaatit	kirjaukset esitetään paikkatietosovelluksen karttanäkymässä
POST	lisää paikkatietokantaan uuden dokumentin	luodaan tietokannan kirjausta vastaava dokumentti paikkatietokantaan
PUT	päivittää yhden tai useamman paikkatietokannassa olevan dokumentin sen uniikin tunnisteen perusteella	käytetään esimerkiksi paikkatietosovelluksen sijainninsyöttö -moduulissa, kun valitulle kirjaukselle lisätään sijaintitieto
DELETE	poistaa yhden paikkatietokannan dokumentin uniikin tunnisteen perusteella	käytetään esimerkiksi paikkatietokannan dokumentin poistamiseen, kun vastaavaa dokumenttia ei enää esiinny CRM-järjestelmän tietokannassa

### 3.3 Tietokantojen yhdistäminen

Kehitysprojektin pilottivaiheessa projektit olivat esitetty kartalla Googlen MyMaps-selainsovelluksessa. MyMaps-kartan taustalla on myös tietokanta, joka voidaan viedä ulos MyMapsista KML- tai KMZ-tiedostoformaattissa (12). KML-formaatin muotoilu noudattaa selkokielistä XML-rakennetta, jolloin karttamerkkien sijainteja ja niille kirjattuja tietoja voidaan tarkastella sekä muuttaa esimerkiksi Windowsin Notepadilla tai vastaavalla tekstinkäsittely sovelluksella. KMZ-formaatissa tiedostokoko on huomattavasti pienempi verrattuna identtiseen KML-tiedostoon, sillä formaatin Z-kirjain viittaa pakattuun muotoon (engl. *zipped*). KMZ-formaatti olisi hyödyllinen siirrettäessä kartan sisältöä toiseen karttaliittymään, mutta tietoja käsin siirrettäessä selkokielineen KML-formaatti osoittautui paremmaksi vaihtoehdoksi.

Pilottivaiheessa kirjattuja tietoja sekä karttamerkkien sijainteja ei tarvitsisi tulkita uudelleen, jos KML-tiedoston sisältö saataisiin liitettyä suoraan osaksi paikkatietokantaa. KML-tiedostosta pystyttiin poimimaan kirjausten avaintiedot Excel-taulukkoon ja muotoilemaan taulukko vastaamaan paikkatietokannan dokumenttien muotoilua (liite 3). Excel-taulukon tiedot tuli viedä edelleen JSON- tai pilkuin eroteltuina tietoina CSV-formaatissa (engl. *Comma Separated Values*), sillä ne ovat ainoita tiedonvaihtoformaatteja, joita MongoDB Compass Community -sovelluksen käyttöliittymä pystyy tulkitsemaan dokumenttien sisään lukemiseksi (13).

Pilottivaiheen kartan tietokannan tuonti paikkatietokantaan onnistui, mutta MyMaps-tietokannan kirjausten kuvaukset eivät noudattaneet mitään ennalta määritettyä muotoilua. Kirjausten kuvaukset sisälsivät lisätietoja kohteen luonteesta, ja tiedot tulisi määritellä uudelleen tuoduille kirjauksille. Pilottivaiheen kartan tiedot voitaisiin siis käsin muotoilla vastaamaan paikkatietokannan dokumenttien skeemaa. Tietokantaintegraatiossa on kuitenkin syvempi ongelma, jossa liitettävän tietokannan kirjauksia ei voida helposti yhdistää vastaavaan CRM-järjestelmän kirjaukseen. Liitettäville kirjauksille tulisi löytyä CRM-järjestelmästä kirjausta vastaava UUID, vaikka ei ole taetta, että kirjausta olisi edes olemassa CRM-järjestelmän tietokannassa. Yhteensopivuuskysymys johtaa siihen, että paikkatietokantaan tuotavat kirjaukset tulisi käsitellä manuaalisesti ja yhdistää CRM-järjestelmän UUID:t kirjauksiin tietojen käsittelyn Excel-muokausvaiheessa tai kirjata UUID suoraan paikkatietokantaan MongoDB Compass Community -sovelluksen avulla. Kolmas mahdollisuus olisi luoda paikkatietosovellukseen työkalu kyseistä prosessia varten, mutta työkalun käyttöaste käyttö-tarkoituksessaan jäisi hyvin pieneksi ja yhden käyttökierroksen jälkeen työkalu olisi lähes hyödytön. Sellaisen työkalun toteutus ei ole kannattavaa, joten siihen ei lähtökohtaisesti käytetä resursseja.

Tietokantaintegraatio todettiin mahdolliseksi, ja MyMaps-tietokannan kirjausten sijaintitiedot ovat jo yhdistettyinä projektien nimimuotoiseen tunnisteeseen. Tietojen yhdistäminen CRM-järjestelmän kanssa voitaisiin ohittaa kokonaan, jolloin tiedot olisi mahdollista esittää väliaikaisesti kartalla rinnakkaisesta

tietokannasta. Rinnakkaisessa tietokannassa kirjaukset olisivat staattisia, sillä ne eivät päivittyisi, jos CRM-järjestelmän tietoja muutetaan. Staattinen esitystapa voisi kuitenkin toimia esimerkiksi tilanteessa, jossa kartalla haluttaisiin väliaikaisesti esittää toisen yrityksen projektitietoja. CRM-järjestelmään MyMaps-tietokannan tiedot kannattaa todennäköisimmin tuoda manuaalisesti Excel-taulukon kautta, kunhan tiedot eheytetään ja mahdollinen tietokantoja yhdistävä UUID on poimittavissa CRM-järjestelmästä.

## 4 Sovelluksen osiot

Paikkatietosovelluksen toteutuksessa on käytetty modulaarista logiikkaa, eli sovellus koostuu osioista, jotka kattavat tietyn toiminnallisuuskokonaisuuden. Modulaarisen toteutustavan kantavaidea on, että osioita voidaan jatkossa vapaasti kehittää lisää ja niiden lisääminen vaatii maksimissaan minimaalisia muutoksia jo julkaistuihin osioihin. Sovelluksen käynnistyksen jälkeen käyttäjille esitetään listaus sovelluksessa käytössä olevista osioista, joita nykyisessä kokoonpanossa ovat karttanäkymä ja sijainnin syöttö.

### 4.1 Käynnistys

Kehitysvaiheessa paikkatietosovellus ja paikkatietokanta tuli erikseen käynnistää komentokehoteella, mutta paikkatietojärjestelmän lopullisessa kokoonpanossa sovelluksen on tarkoitus avautua tavallisen työpöytäsovelluksen tavoin. Sitä varten selvitettiin, että Electron-sovellus voidaan paketoita exe-asennustiedostoksi esimerkiksi Electron packagerilla (14).

Paikkatietosovelluksen käynnistyksessä suoritetaan koko järjestelmän oleellimmat tiedon tulkinta ja -käsittelytoiminnallisuudet. Keskeisin ja paikkatietoteknisesti tärkein niistä on automaattisesti tapahtuva koordinaattien tulkinta, joka perustuu käyttäjän CRM-järjestelmään syöttämiin osoite tai koordinaattitietoihin. Käynnistysmoduuli yrittää ensisijaisesti muuttaa kirjauksen kuvauksessa annetun lähiosoitteen Google Geocoder -ohjelmointirajapinnan avulla WGS84-koordinaateiksi, joiden koordinaatistojen tunniste, eli EPSG-koodi on 3857 (15).

Muutetut tai jo oikeassa koordinaatistossa syötetyt koordinaatit tallennetaan paikkatietokantaan kohdekirjausta vastaavalle dokumentille. Googlen geokoodauksen rajapinta tunnistaa syötettyjen koordinaattien muodon, joten koordinaatit eivät voi päätyä paikkatietokantaan väärässä koordinaatistossa.

Paikkatietosovelluksen automaattisesti suorittamasta sijaintitietojen käsittelystä jätetään CRM-järjestelmään kyseiselle kirjaukselle tietokantamerkintä käsittelyn tilasta taulukon 2 mukaisesti.

Taulukko 2. Tietokantamerkintöjen toimintakuvaukset.

Merkintä	Toimintakuvaus
m1	Paikkatietosovellus on käynnistyksen yhteydessä lukenut käyttäjän CRM-järjestelmän kirjaukselle syöttämät tiedot, eikä sijainnin määrittäminen ole niiden perusteella onnistunut tai tekstikenttä on jätetty tyhjäksi. Kirjauksen sijainti on siirretty paikkatietosovelluksen sijainnin syöttömoduulilla käyttäjän tulkittavaksi.
m2	Käyttäjä on sijainnin syöttömoduulilla liittänyt sijaintitiedon kirjaukselle, jonka tietokantamerkintä on ollut m1.
m3	Paikkatietosovellus on käynnistyksen yhteydessä lukenut käyttäjän CRM-järjestelmän kirjaukselle syöttämät tiedot ja tulkinnut niiden perusteella kohteelle sijaintitiedon onnistuneesti.

CRM-järjestelmän kirjauksen kuvaukseen lisättävässä tietokantamerkinnässä on esitetty tietoja sijainnin käsittelyn tilasta ja merkinnässä ohjeistetaan, kuinka käyttäjä voi halutessaan nollata kirjaukselle tulkitun sijaintitiedon (taulukko 3). Tietokantamerkintä avustaa myös kehitystiimiä selvittämään paikkatiedon käsittelyprosessin toimintaa tilanteissa, joissa sijaintitiedon tulkinta ei ole toiminut odotetusti.

Taulukko 3. CRM-järjestelmän kirjaukselle jätettävän tietokantamerkinnän muotoilu.

Merkintä	Tietokantaan kirjoitettava merkintä
m1	Projektille ei ole tallennettu sijaintitietoa. Paikkatietosovellus on automaattisesti tulkinnut kirjauksen sisällön, joten älä muuta tämän merkinnän muotoilua. Mikäli haluat syöttää projektille uuden sijainnin, tyhjennä tämä tekstikenttä kokonaisuudessaan ja syötä uusi sijainti lähiosoitteena tai WGS84-koordinaatitena. Voit myös jättää tekstikentän tyhjäksi, jolloin sijainti siirtyy syötettäväksi

	paikkatietosovelluksen paikkatietomoduuliin. Virhetilanteissa olkaa yhteydessä ylläpitoon. Dev code: m1
m2	Käyttäjän paikkatietomoduulilla tallentama sijainti on: <i>WGS84-koordinaatit, tulkittu lähiosoite ja kunta</i> . Älä muuta tämän merkinnän muotoilua. Tyhjennä tämä tekstikenttä kokonaisuudessaan, jos projektin sijaintitiedoissa on virhe. Mikäli projektin sijaintitiedoissa on virhe, tyhjennä tämä tekstikenttä kokonaisuudessaan. Uuden sijainnin voit liittää kirjoittamalla tyhjennettyyn tekstikenttään lähiosoitteen tai WGS84-koordinaatit. Muussa tapauksessa jätä tekstikenttä tyhjäksi, ja projekti siirtyy paikkatietosovelluksen paikkatietomoduuliin odottamaan sijainnin määrittystä. Virhetilanteissa olkaa yhteydessä ylläpitoon. Dev code: m2
m3	Projektille on automaattisesti tulkittu sijainti, jonka lähiosoite on: <i>osoite, kunta</i> . Älä muuta tämän merkinnän muotoilua. Mikäli automaattisesti tulkittussa sijaintitiedossa on virhe, tyhjennä tämä tekstikenttä kokonaisuudessaan. Uuden sijainnin voit liittää kirjoittamalla tyhjennettyyn tekstikenttään uuden lähiosoitteen tai WGS84-koordinaatit. Paikkatietosovellus muuttaa annetut tiedot sijaintitiedoksi käynnistyksen yhteydessä. Muussa tapauksessa jätä tekstikenttä tyhjäksi, ja projekti siirtyy paikkatietosovelluksen paikkatietomoduuliin odottamaan sijainnin määrittystä. Virhetilanteissa olkaa yhteydessä ylläpitoon. Dev code: m3

Tietokantamerkintöjen ihmiselle osoitetun tiedon lisäksi paikkatietosovellus tunnistaa merkintöjen avulla, että jo merkityille kirjauksille ei ole syytä suorittaa tiedonkäsittely- ja paikkatiedontulkintatoiminnallisuuksia uudelleen. Merkityt kirjat on jo aiemmin tulkittu, ja siksi ne voidaan rajata käynnistyksen tiedonkäsittelyn ulkopuolelle heti kättelyssä. Näin vältytään turhalta tietojenkäsittelyltä ja onnistutaan vähentämään sovelluksen työtaakkaa, joka mahdollisesti vaikuttaisi sovelluksen käynnistysaikaan.

Paikkatietojärjestelmän toimintaa ei ole testattu suurella yli tuhannen kirjauksen määrällä, jonka käsittelystä todennäköisesti ei synny edes merkittävää odotusaikaa. Turhalla kirjausten uudelleenkäsittelyllä kirjausten tietosisällöt asetetaan alttiiksi riskeille, sillä esimerkiksi tietokoneen virran katketessa sovelluksen suoritus keskeytyy, jolloin toiminnallisuuksien loppuun saattamisesta ei ole takuuta. Keskeytyksen seurauksena kirjauksille voisi päätyä vääriä tai keskenäisiä tietoja, joiden paikallistaminen ja tarkastaminen olisi työlästä. Sijaintitietojen turha käsittely aiheuttaisi myös ylimääräisiä pyyntöjä Googlen geokoodauspalvelulle, jolloin syntyisi lisäkustannuksia. (22)

Sijainnin tulinnan ja CRM-järjestelmän tietokantamerkkintöjen käsittelyn jälkeen tietokantojen välillä suoritetaan tarkastuksia niiden eheyden ja vastaavuuden ylläpitämiseksi. Esimerkiksi paikkatietokannasta poistetaan kirjaukset, jotka eivät enää esiinny CRM-järjestelmässä. Tilanteeseen voidaan päätyä, jos CRM-järjestelmästä kirjaus poistetaan kokonaan tai sen tila palautetaan tarjouskäsitelyyn. Oleellisia tietoja ei menetetä poistamisen tai palautuksen yhteydessä, sillä poistettavilla kirjauksilla ei oletettavasti ole säilytettäviä tietoja ja projektien palautuessa tarjouskäsitelyyn kirjauksille ei ole lisätty tarkentavia tietoja. Eheydeltä tarkastuksessa sovellus vertailee tietokantojen UUID-taulukkoja keskenään (esimerkkikoodi 3), jonka tuloksena jäljelle jääneitä uniikkeja tunnuksia vastaavat dokumentit poistetaan paikkatietokannasta (esimerkkikoodi 4).

```
const nonExistantEntriesToBeDeleted = _.difference(allMiddleDbIds,
allDbIds)
```

**Esimerkkikoodi 3.** Difference-metodilla paikkatietokannan tunnuksien taulukosta karsitaan pois kaikki CRM-järjestelmän tietokannassa esiintyvät tunnuksien dokumentit pois. Uuden muuttujan taulukkoon jääneiden tunnuksien mukaiset dokumentit poistetaan paikkatietokannasta.

```
const deleteFromMiddleDatabase = (item) => {
  return new Promise((resolve) => {
    var deleteSettings = {
      "url": api/project/location + item,
      "method": "delete",
      "timeout": 0,
    };
    $.ajax(deleteSettings).done(response => {
      resolve(console.log("deleted " + response.length + " documents from middle-database"))
    })
  })
}

async function deleteNonExistantEntriesFromDatabase() {
  nonExistantEntriesToBeDeleted.forEach(async item => {
    await deleteFromMiddleDatabase(item)
  })
}
```

**Esimerkkikoodi 4.** Funktio, jolla paikkatietokannasta poistettavien kirjausten taulukon kohteet poistetaan yksi dokumentti kerrallaan.

## 4.2 Kartta

### 4.2.1 Karttanäkymä

Paikkatietokannan jokaisella dokumentilla on pistetyyppinen geometriaobjekti (liite 1), jonka taulukkoon koordinaatit ovat tallennettuina. Paikkatietosovellus hakee GET-pyyntöillä paikkatietokannan ohjelmointirajapinnasta ne kirjaukset, joilla on hyväksytyt koordinaatit (esimerkkikoodi 5). Kirjausten tietojen perusteella sovellus luo kartalle karttamerkkiobjektit ja sijoittaa ne paikoilleen (esimerkkikoodi 6).

```
const coordEntries = () => {
  return new Promise((resolve, reject) => {
    let settings = {
      "url": "/projektit/sijainti",
      "method": "GET"
    }
    $.ajax(settings).done(res => {
      let result = res.map(item => {
        return item
      })
      return resolve(result)
    })
  })
}
```

Esimerkkikoodi 5. Paikkatietokannan ohjelmointirajapinnan päätepisteessä */projektit/sijainti* on paljastettu kaikki projektit, joille on osoitettu sijaintitiedot. Paikkatietosovellus hakee päätepisteestä kaikki kirjat Jquery ajax GET-metodilla.

```

const allCoordEntries = await coordEntries()

async function markerHandle() {
  allCoordEntries.forEach(async item => {
    await addMarker(item).then(marker => {
      gmapsMarkers.push(marker)

      clickEvent(marker, item, keywords)
    })
  })
}

const addMarker = (item) => {
  return new Promise((resolve, reject) => {
    let status = item.job_status

    let lat = item.geometry.coordinates[0]
    let lng = item.geometry.coordinates[1]
    let location = new google.maps.LatLng(lat, lng)

    let marker = new google.maps.Marker({
      title: item.job_number.toString(),
      position: location,
      category: status,
      map: map,
      icon: icon,
      description: item.job_name,
      id: item._id
    })
    resolve(marker)
  })
}

```

**Esimerkkikoodi 6.** Karttamerkit luodaan ja niille poimitaan paikkatietokannan dokumentilta mm. nimike, numero, sijainti, tila ja UUID. Valmiit karttamerkkiobjektit lisätään valmiiden karttamerkkien *gmapsMarkers*-taulukkoon.

Visuaalista paikkatiedon analysointia varten karttanäkymän yläreunaan toteutettiin suodattimien pudotusvalikko, josta käyttäjä voi vaihtaa karttamerkkien esitystavan vastaamaan käynnissä olevia ja päättyneitä projekteja. Paikkatietosovelluksen jatkokehityksessä suodattimia on tarkoitus luoda lisää ja samalla mahdollistaa, että useampi kuin yksi suodatin voi olla samanaikaisesti aktiivisena. Käyttäjät ovat toivoneet lisäsuodattimiksi mm. projektien laajuutta kuvaavaa arvoa sekä mahdollisuutta rajata kohteita esitettäväksi niiden tyyppin perusteella. Lisäsuodattimilla kartalla olisi mahdollista esittää esimerkiksi kaikki suuret koulurakennushankkeet, joilla yritys on ollut mukana.

Karttanäkymässä suodattimien oikealla puolella on, jonka tarkoituksena on helpottaa navigointia karttanäkymässä. Käyttäjä voi hakutyökaluilla löytää

yksittäisen projektin sen numeron perusteella tai kohdentaa näkymän osoitteeseen tai muuhun Google Mapsiin merkittyyn kiinnostuksen kohteeseen. Mapsin hakutoiminnallisuudet rajattiin toimimaan yrityksen ydintoiminta-alueelle, sillä muuten paikkojen nimiä haettaessa kartta saattaisi siirtyä kohteeseen, joka ei sijaitse Suomessa.

Hakutyökalu toteutettiin siten, että sen tekstinsyöttökentässä on tapahtuma-kuuntelija (engl. *Event Listener*), joka poimii käyttäjän syöttämän merkkijonon hakutermitse. Ensimmäisenä paikkatietosovellus tarkastaa, sisältääkö syötetty merkkijono pelkästään numeroita, jolloin merkkijono todetaan projektinumeroksi. Vahvistuksen jälkeen syötettyä hakutermiä verrataan kaikkien karttanäkymässä olevien karttamerkkiobjektien projektinumeroiden kanssa ja osuman löydyttyä näkymä keskitetään kyseisen karttamerkin koordinaatteihin. Tapauksessa, jossa syötetty merkkijono sisältää kirjaimia, hakutermi todetaan osoitteeksi tai paikan nimeksi. Silloin hakutermiä yritetään geokoodata geokoodaus-funktiolla, joka hyödyntää Googlen Geocoder-ohjelmointirajapintaa (esimerkkikoodi 7). Geokoodauksen onnistuessa funktio palauttaa Google Geocoderin tulkitsemat koordinaattitiedot, joihin karttanäkymä keskitetään. Geokoodauksen epäonnistuessa funktiosta palautuu käyttäjälle vaste, jossa kerrotaan geokoodauksen virheestä.

```

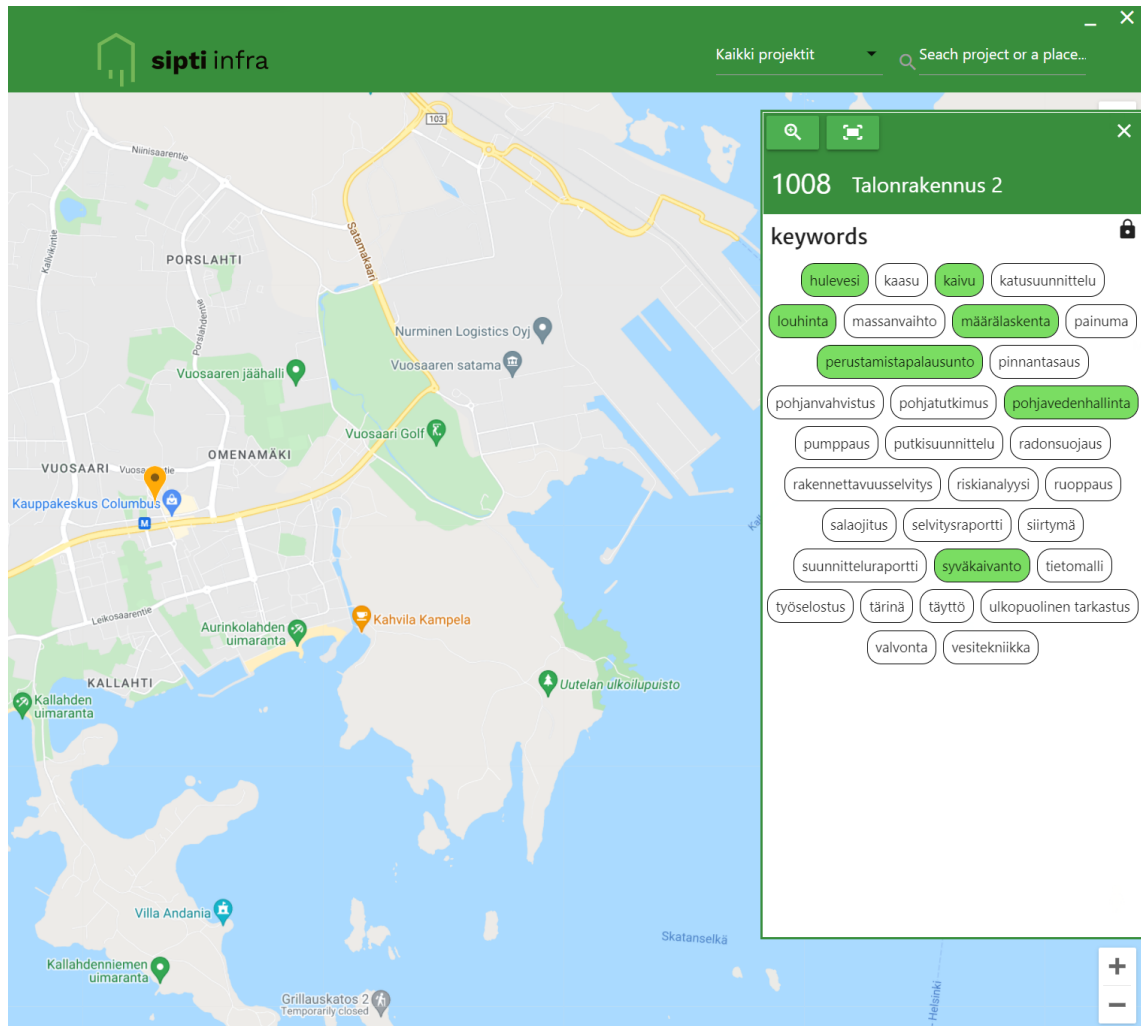
const geocodeAddress = (searchTerm) => {
  return new Promise((resolve) => {
    geocoder.geocode({"address": searchTerm + " finland"}, (re-
      sults, status => {
        if (status == "OK") {
          function show() {
            map.setCenter(results[0].geometry.location)
            map.setZoom(14)
          } resolve(show())
        } else {
          M.toast({
            html: "Osoite tai paikka on virheellinen!",
            displayLength: 4000,
            inDuration: 1000,
            outDuration: 1000
          })
        }
      })
    })
  })
}

```

Esimerkkikoodi 7. Syötetty hakutermin siirretään geokoodaus-funktioon. Geokoodauksen onnistuessa karttanäkymä keskitetään tulkittuun sijaintiin, ja jos sijaintia ei ole olemassa, käyttäjälle annetaan siitä vaste.

#### 4.2.2 Tietoikkuna

Karttamerkkiä painettaessa näkymän oikeaan laitaan avautuu tietoikkuna, johon on poimittu valitun projektin avaintietoja. Kehitysvaiheen sovellusversion tietoikkunassa on esitetty projektin numero, nimi ja kirjaukselle yhdistetyt tunnisteet (kuva 4). Avaintiedoista numero ja nimi haetaan suoraan karttamerkkiobjektilta, kun taas kirjauksen tunnisteet ja niiden aktiivisuuden tilan tiedot tulevat verkon ylitse CRM-järjestelmän ohjelmointirajapinnasta.



Kuva 4. Karttanäkymän tietokkunanassa on esitetty kohteen tietoja sekä avainsanoja.

Tietokkunaan voidaan tietokannoista poimia enemmänkin tietoja esitettäväksi, ja käyttäjät toivoivatkin, että sovelluksen seuraavan version tietokkunanassa näkyisi jo esitettyjen tietojen lisäksi asiakkaan nimi, toteutuksen aikaväli ja projektin laajuutta kuvaava tieto. Valmiiden tietojen esittäminen paikkatietosovelluksessa on yksinkertaista, mutta toteutuksen aikavälin ja laajuuden tiedot vaativat tarkemmin pohdittuja arvoja.

Projektin toteutusajan, eli aloituksen ja päättymisen aikavälin määrittäminen ns. putoaa jo olemassa olevien yksiselitteisten tietojen esittämisen ja kokonaan uuden tiedon luonnin välimaastoon. Tieto aloituksesta ja päättymisestä kirjautuu automaattisesti CRM-järjestelmään perustuen kirjauksen luonnin ja projektin

sulkemisen ajankohtiin. Ajankohtien aikaleimatiedot ovat usein vääristyneitä, joten niitä ei voida suoraan tulkita toteutuksen aikaväliksi. Projektit nimittäin harvoin varsinaisesti alkavat kirjauksen luonnin tai sen tilan muutoksen yhteydessä, eikä kirjauksia aina suljeta projektin virallisena päättymishetkenä. Satunnaisesti kirjauksia jää CRM-järjestelmään avoimiksi, vaikka niillä tehty työ on päättynyt. Siksi kirjauksia on aiemmin suljettu kollektiivisesti tiettyä ajankohdana, mutta silloinkin lopetusajankohta vääristyy. Järkevän toteutusaikavälin määrittämisessä onnistuttaisiin sisäisellä ohjeistuksella ja koulutuksella, jossa käyttäjiä opastettaisiin jatkossa sulkemaan projektit niiden todellisena päättymishetkenä. Toteutusajan kirjaamisen tarkkuudeksi on valittu vuosi, sillä se on riittävä tarkkuustaso ja kollektiivista sulkemistapaa voidaan soveltaa tarvittaessa. Sen ajankohta tulee määrittää säännöllisesti tehtäväksi ennen vuodenvaihdetta, jolloin kirjautuvassa päättymisen aikaleimassa on oikea vuosiluku.

Yksittäisen projektin laajuuden määrittämisessä on pohdittava, mikä oikeasti olisi laajuutta kuvaava arvo. Sitä on mahdollista yksinkertaisimmillaan kuvata myynnin arvolla tai resurssitoteumalla. Kumpikaan niistä ei anna täyttä kuvaa projektin laajuudesta, sillä myyntiarvoperusteisesti kannattavat ja resurssiperusteisesti kannattamattomat projektit yli korostuisivat otannassa. Kehitysvaiheessa on ehdotettu, että toteutuneen laajuuden määreen voisi ratkaista laskukaavalla, joka yhdistää myyntiarvon ja resurssitoteuman keskenään. Niistä johdettu arvo muutetaan vastaamaan laajuutta kuvaavaa sanaa, joita ovat mikro, pieni, keskikokoinen, merkittävä, suuri ja erittäin suuri.

Alustavasti ehdotettu projektin toteutuneen laajuuden ( $PL$ ) ehdotettu laskukaava on esitetty kaavassa 1, jossa  $e$  on kokonaismyynnin arvo euroina ja  $Rt$  on resurssitoteuma tunteina.

$$PL = \frac{e}{100} + Rt \quad (1)$$

Esitetyllä laskukaavalla projektin toteutuneen laajuuden arvo ei ole ainoastaan riippuvainen toisesta avainarvosta, vaan niistä kumpikin huomioidaan. Esimerkitapauksessa myyntiarvon ollessa nolla ja resurssitoteuman kymmenen

laajuudeksi saadaan 50 % projektin laajuuden maksimiarvosta. Lasketun arvon tarkoitus ei ole kuvata kannattavuutta, sillä CRM-järjestelmässä on sisäänrakennettuna omia keskeisiä suorituskykyindikaattoreita (*KPIs*, engl. *Key Performance Indicators*), joilla kannattavuutta seurataan projektitasolla (16).

### 4.3 Tunnisteet

CRM-järjestelmälle voidaan tuottaa lisäarvoa liittämällä kirjauksille tunnisteita. Niitä olisi mahdollista hyödyntää esimerkiksi paikkatietosovelluksen karttanäytymän suodatinkriteereinä, järjestelmien hakutoimintojen tehostamisessa ja raportointiominaisuuksissa. Aiemmin tunnisteiden käyttö ei ole ollut merkittävää projektienhallinnassa, eikä käytössä ole ollut vakioituja tunnistelijoita. Alhainen käyttöaste toimii kehitysprojektin eduksi, sillä tunnistelijoita muutettaessa CRM-järjestelmän kirjauksilta ei katoa oleellista tietoa.

Tunnisteilla on tarkoitus kuvata projektilla tehtyä työtä avainsanojin, joita olisi käytössä noin kaksikymmentä. Avainsanojen lisäksi projektit voidaan tunnistaa ylempällä tasolla projektin kohdetta kuvaavalla nimikkeellä. Toista käyttöön otettavaa tunnistetta kutsuttaisiin kategoriaksi, ja avainsanoista poiketen kirjauksella voi olla ainoastaan yksi kategoria aktiivisena. Valittavissa olisi yhteensä noin kymmenen kategoriaa, joista esimerkkeinä on asuinrakennustuotanto, julkinen sekä toimitilarakentaminen.

Tunnistelistan ajantasainen muotoilu haetaan CRM-järjestelmästä kartan latautumisen yhteydessä. Avainsanojen listaa tai niiden aktiivisuuden tilaa ei ole kannattavaa muiden tietojen tapaan tallentaa karttamerkille, jotta kartta latautuisi nopeammin ja CRM-järjestelmän rajapintaan lähetettävien pyyntöjen määrä pidettäisiin kurissa. Karttamerkkeihin on liitetty klikkaustapahtumakuuntelija, joka hakee näkyvässä valitun kirjauksen tunnisteiden tilan CRM-järjestelmästä GET-pyyntöllä ja muotoilee avainsanat tietoiikkunaan esitettäväksi (esimerkkikoodi 8).

```

const clickEvent = (marker, item, keywords) => {
  google.maps.event.addListener(marker, "click", () => {
    getTags(marker, item, keywords).then(val => {
      $("#tagcontainer").html(val)
    })
  })
}
const getTags = (marker, item, keywords) => {
  return new Promise((resolve) => {
    var settings = {
      "url": "https://api.CRM.com/version/endpoint" +
        marker.id,
      "method": "GET",
      "timeout": 0,
      "headers": {
        "ClientID": process.env.CLIENT_ID,
        "Authorization": bearerToken,
      }
    };
    $.ajax(settings).done(() => {
      const existingTags = item.job_tags
      function formTags() {
        let tags = []
        existingTags?.forEach(item => {
          let targetObject = {name: item, value: true}
          tags.push(targetObject)
        }); return tags
      }
      const mapTags = formTags()
      const combinedTags = _.unionBy(mapTags, keywords,
        "name")
      const allTags = _.sortBy(combinedTags, (a) => {return
        a.name})
      function tagstate() {
        return new Promise((resolve) => {
          const allDivs = _.map(allTags, item => {
            if (item.value !== null) {
              let newDiv = document.createEl-
                ement("div")
              newDiv.className = "tag-active"
              let tagElement = document.createEl-
                ement("span")
              tagElement.textContent = item.name
              newDiv.appendChild(tagElement)
              return newDiv
            } else if (item.value === null) {
              let newDiv = document.createEl-
                ement("div")
              newDiv.className = "tag-inactive"
              let tagElement = document.createEl-
                ement("span")
              tagElement.textContent = item.name
              newDiv.appendChild(tagElement)
              return newDiv
            }
          })
          resolve(allDivs)
        })
      }
      resolve (tagState())
    });
  });
}
}

```

Esimerkkikoodi 8. Karttamerkin painalluksen tapahtumakuuntelijassa on avainsanojen hakufunktio, joka luo CRM-järjestelmässä määritellyn tunnistelistan esitettäväksi tietokkunaan ja muotoilee tunnisteet niiden aktiivisuuden tilan perusteella.

Tietokkunassa avainsanat on pyritty esittämään mahdollisimman helppolukuisina, sillä avainsanoja on paljon ja niistä käyttäjän tulisi löytää etsimänsä tunniste nopeasti. Avainsanat on aakkosjärjestelty ja niitä on esitetty vierekkäin niin monta kuin ikkunan riville mahtuu, sillä ihminen prosessoi yhdellä silmäyksellä vaakasuuntaisia listauksia jopa kaksi kertaa paremmin kuin vastaavia pystysuuntaisia listauksia (17). Tietokkunassa avainsanat on keskitetty, ja rivin täytyessä seuraava avainsana siirtyy uudelle riville. Näin tunnistelistan muokkaamisenkin jälkeen avainsanat asettuvat tietokkunassa järkevästi paikoilleen.

Tunnistelistaa muokataan CRM-järjestelmän käyttöliittymässä, ja siellä listaan voi avainsanoja lisätä, muuttaa, poistaa käytöstä tai poistaa kokonaan. Avainsanan luonnin jälkeen se on heti liitettävissä kirjaukselle ja liittämisen jälkeen se tulkitaan kirjauksella aktiiviseksi. Poistamalla avainsana kirjaukselta tunniste tulkitaan inaktiiviseksi.

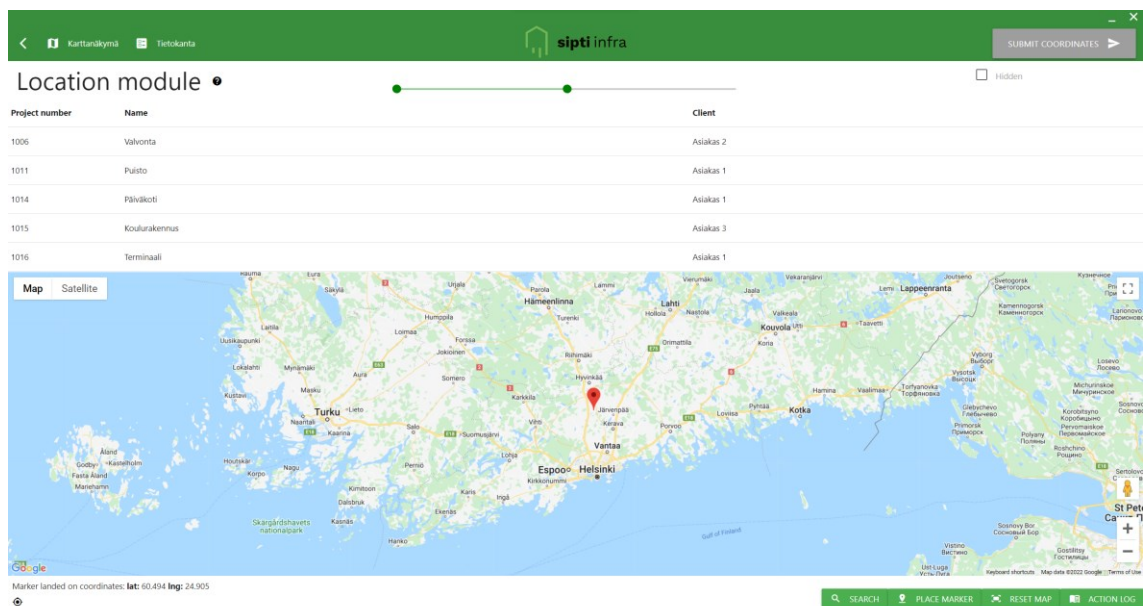
Avainsanojen aktiivisuuden hallinnointia varten karttanäkymään kehitettiin vaihtoehtoinen toimintatapa, jotta käyttäjä voi karttaselailun lomassa muuttaa aktiivisia avainsanoja valitsemallaan kirjauksella. Soveltamalla uutta toimintatapaa käyttäjän ei tarvitse avainsanoja muokataksaan etsiä valitsemaansa kohdetta CRM-järjestelmästä, vaan tietokkunassa avainsanat ovat reaaliaikaisesti muokattavissa. Käyttäjän tulee ainoastaan vapauttaa avainsanojen muokkaustoiminto käyttöön avaamalla tietokkunan oikeassa yläkulmassa sijaitseva lukitus-painike, minkä jälkeen avainsanojen aktiivisuuden tilat ovat vaihdettavissa. Avaustoiminnolla minimoidaan virheitä, joita käyttäjä saattaisi epähuomiossa tehdä esimerkiksi vahingossa painamalla avainsanaa.

Tunnisteiden uutta hallinnointitapaa ei pystytty kehitysvaiheessa viimeistelemään, sillä CRM-järjestelmän ohjelmointirajapinnassa ei tuolloin ollut URL-päätepistettä avainsanojen hallinnoinnin pyynnöille. Toiminnallisuus on kuitenkin toteutettu mahdollisimman pitkälle, ja avainsanojen päätepisteen tilannetta

tarkastellaan uudelleen käyttöönottovaiheessa ja samalla toiminto voidaan viimeistellä.

#### 4.4 Sijainnin syöttö

Projekteille maantieteellinen sijainti tulkitaan ensisijaisesti automaattisesti kirjaukselle syötetyn tiedon perusteella. Tiedon puuttuessa tai sijainnin tulkinnan epäonnistuessa kirjaukselle jätetään tietokantamerkintä m1, jolla viitataan siihen, että kirjaus odottaa sijaintitietoa manuaalisesti syötettäväksi. Sitä varten paikkatietosovellukseen toteutettiin sijainnin syöttömoduuli, jonka näkymään on listattu kaikki sijainnittomat m1-merkityt kirjat pienin tunnustenumero ensimmäisenä. Käyttäjälistasta voi valita tuntemansa projektin, siirtää karttamerkin hankkeen sijaintiin ja tallentaa sijaintitiedot vain muutamalla klikkauksella. Sijainnin syötön vakionäkymä on esitetty kuvassa 5.



Kuva 5. Sijainnin syötön vakionäkymä, jossa käyttäjä voi halutessaan osoittaa tietokantakirjaukselle koordinaatit sijainnin syöttötoiminnolla.

Sijainnin syötön näkymässä kartta esitetään mahdollisimman suurena, jotta karttamerkin siirtäminen haluttuun sijaintiin olisi mahdollisimman selkeää. Loput tilasta käytetään sijainnittomien kirjausten listaukselle, joka saa viedä

maksimissaan noin puolet näkymän tilasta, jotta kartta säilyy tulkittavana. Kirjausten listaus on tarkoitettu ajateltavaksi aikajanana, jota navigoidaan manipuloitavalla aikajanalla (kuvassa 5 keskellä ylhäällä). Esimerkkinä kuvan 5 listauksessa on esitetty viisi kirjausta, ja projektijanaa kokonaisuutena oikealle siirtämällä listaukseen vaihtuvat näytettäväksi ne viisi kirjausta, joiden kohdalle projektijana asetetaan. Projektijanan pituus on muokattavissa kummastakin suunnasta, jotta käyttäjä voi muuttaa kartan kokoa suuremmaksi. Janaa muokkaamalla listassa voi minimissään esittää yhden kirjauksen ja maksimissaan sovelluksen ikkunan kokoon perustuvan rajoittimen mukaisen maksimimäärän.

Kehityksen myöhemmässä vaiheessa loppukäyttäjältä saatiin kehitysehdotus, jonka mukaan listaus ja kartta voisivat näkymässä olla rinnakkain. Se on toimiva idea, sillä Suomi on pidempi pohjois-eteläsuuntaisesti, jolloin kartassa näkyisi enemmän relevanttia aluetta ja kirjausten listaus voisi olla pidempi. Sijainnin syöttömoduulin näkymän komponenttien asettelua parannellaan saadun palautteen perusteella jatkokehitysvaiheessa.

Yksittäiselle kirjaukselle sijaintitiedon liittäminen onnistuu yhtä lailla sekä sijainnin syöttömoduulilla, että CRM-järjestelmän kautta. Sijainninsyöttömoduulin täysi potentiaali tulee esille, kun suurelle määrälle kirjauksia osoitetaan sijaintitiedot järjestelmällisesti. CRM-järjestelmän käyttöliittymässä ei ole mahdollista rajata näkyviin ainoastaan niitä kirjauksia, joilla ei ole sijaintia, joten sijainnittoimien kirjausten listaaminen joukkona sijainnin syöttömoduulissa nopeuttaa paikkatiedon liittämisooperaatiota. Pienellä sijainninsyöttömoduulin koodin muutoksella ja valitsimen toteutuksella olisi myös mahdollista osoittaa koordinaatteja toisen paikkatietokannan kirjauksille, mikäli siihen on tarvetta.

Käytännössä sijainnin syöttäminen kirjausten listasta valitulle kohteelle tapahtuu siirtämällä karttamerkki paikoilleen ja hyväksymällä sen koordinaatit kuvan 5 oikeassa ylälaudassa olevalla napilla. Käyttäjä voi tehdä kirjauksen valinnan ja karttamerkin siirtämisen toiminnot haluamassaan järjestyksessä, sillä sovellus tarkastaa ennen sijaintitiedon tallentamista, että kummatkin toiminnot on suoritettu ja sen indikaattorina nappi muuttuu harmaasta vihreäksi. Karttanäkymän

alareunaan (kuva 5) on lisätty muutama toiminto kartan ja sen merkin manipuloimista helpottamaan. Kartan näkymän voi esimerkiksi siirtää hakutoiminnolla osoitteeseen tai paikkaan, karttamerkin sijainnin voi nollata näkymän keskelle tai kartan sekä sen merkin voi palauttaa lähtötilanteeseen.

Karttamerkkiä siirrettäessä näkymän vasemmassa alakulmassa (kuva 5) käyttäjälle annetaan tiedoksi karttamerkin tila ja sille asetetut koordinaatit. Karttamerkin tilan alle ilmestyy sijainti-ikoni, kun karttamerkki pudotetaan johonkin sijaintiin kartalla. Sijainti-ikonia painettaessa sovellus käänteisgeokoodaa koordinaatit ja palauttaa käyttäjälle pudotetun karttamerkin sijainnin lähiosoitteena (15). Toiminnon avulla käyttäjä voi varmistua sijainnin oikeellisuudesta ennen koordinaattien tallentamista. Jokainen Google Geocoder ohjelmointirajapintaan toimitettava geokoodauspyyntö on maksullinen, ja toiminnosta aiheutuisi ylimääräisiä kustannuksia, jos pudotetun karttamerkin lähiosoite päivitetäisiin aina merkin siirtämisen jälkeen ja siksi toiminto on tarkoituksella rajoitettu erillisen käskyn taakse (18).

## 5 Käyttäjäkokeilu

Paikkatietojärjestelmän toteutuksen loppupuolella paikkatietosovelluksen ensimmäistä versiota varten suunniteltiin ja toteutettiin käyttäjäkokeilu. Kokeilua varten koottiin loppukäyttäjien joukosta testiryhmä, jonka tehtävänä oli arvioida paikkatietojärjestelmän toiminnallisuutta kokonaisuudessaan. Käyttäjäkokeilun yhteydessä kartoitettiin myös käyttäjien sen hetkistä CRM-järjestelmän omaksumistasoa ja mielipiteitä sekä ideoita sovelluksen jatkokehittämisestä. Kokeilutilaisuus järjestettiin yhdellä työpisteellä, jolla testiryhmän käyttäjät saivat yksi kerrallaan kokeilla paikkatietosovelluksen viimeisintä kokoonpanoa, joka oli liitetty CRM-järjestelmän kehitysympäristöön. Käyttäjäkokeilun tueksi laadittiin Google Formsilla yhdistetty ohjeistus- ja kyselylomake.

CRM-järjestelmän kehitysympäristöön ei saatu kloonattua todellisen CRM-järjestelmän sisältöä, jotta käyttäjäkokeilu olisi voitu toteuttaa todellisella ja käyttäjille tutulla aineistolla. Kehitysympäristöön luotiin kirjaussetti, jonka sisältö ja

muotoilu jäljitteli todellisen ympäristön kirjauksia ja niitä räätälöitiin hieman palvelemaan käyttäjäkokeilun tavoitteita. Käyttäjäkokeilun alussa dokumentilla avattiin tilaisuuden tarkoitusta sekä tavoitteita ja listattiin sovelluksen puutteet ja poikkeamat, jotta käyttäjät eivät välttämättä tarttuisi tiedossa oleviin seikkoihin paikkatietojärjestelmän kokonaisuutta arvioidessaan. Ohjeistus- ja kyselydokumentilla oli selostuksien ja kysymysten lisäksi annettu käyttäjille tehtäviä, jotta käyttäjät osallistuisivat aktiivisesti kokeiluun. Annettuihin tehtäviin kuului mm. sijainnin liittäminen uudelle kuvitteelliselle kirjaukselle, kokeiluympäristön kirjausten manipulointi, karttanäkymässä surffailu ja näkymän toimintojen kokeilu.

Käyttäjät saivat ensimmäiseksi tehtäväkseen liittää sijainnin lähiosoitteen muodossa CRM-järjestelmän kehitysympäristöön luomalleen kirjaukselle. Sovellus paketoidaan käyttöönottovaiheessa, joten kokeilutilaisuudessa käyttäjien tuli poikkeuksellisesti käynnistää sovellus komentokehoteella. Käynnistyttyään sovellus oli muuttanut annetun lähiosoitteen koordinaateiksi, ja kirjaus sijoitettiin automaattisesti kartalle. Käyttäjiä kehoitettiin karttanäkymässä etsimään oma kirjauksensa ja kertomaan, kuinka helppoa sijainnin liittäminen oli, kuinka he löysivät kohteen kartalta ja olisiko se luonnollista löytää jollakin muulla tavalla. Tehtävän ja vastailun jälkeen käyttäjien annettiin vapaasti tutustua karttanäkymän muihin toimintoihin. Seuraavana tehtävänä käyttäjät saivat kokeilla avainsanojen liittämistä suoraan kirjaukselle karttanäkymän tietokunassa, ja sillä selvitettiin, kuinka todennäköisesti käyttäjät muuttaisivat avainsanoja tietokunasta käsin. Yleisesti tietokunan sisältöön liittyen selvitettiin, mitä lisätietoja tietokunassa olisi hyvä esittää.

Aivan kyselyn lopuksi käyttäjille esitettiin paikkatietojärjestelmän ylläpito- ja jatkokehitysvaiheen mahdollisia kehityssuuntia ja kartoitettiin käyttäjien mielipiteitä niistä. Samalla selvitettiin kehitysideoiden toteuttamisen kannattavuutta liittämällä idean yhteyteen erilliseen kysymykseen arviot idean toteuttamiseen käytettävistä resursseista. Näin kehitysideoiden suuntaa pystytään jo alustavasti muokkaamaan käyttäjäkokeilun tulosten perusteella. Kyselylomakkeella

myös kannustettiin käyttäjiä osallistumaan paikkatietosovelluksen kehityksen ideointiin jättämällä omia ideoitansa kehitystiimille.

Kokeilutilaisuus kesti käyttäjästä riippuen noin tunnin verran, ja käyttäjäkokeilun jälkeen loppukäyttäjät olivat innoissaan paikkatietojärjestelmästä. Käyttäjäkokeilu oli siis onnistunut, ja kehitystiimi sai arvokasta palautetta kehityksen tueksi ja kokeilun perusteella pystyttiin toteamaan, että nykyinen paikkatietojärjestelmän kokoonpano on kaikin puolin käyttöönottokelpoinen. Käyttäjäkokeilussa ei tullut odottamattomia tai kriittisiä epäkohtia ilmi.

## **6 Seuraavat askeleet**

Insinööriyön valmistumisen jälkeen kehitysprojektia jatketaan, ja asialistalla on sovelluksen käyttöönoton lisäksi jatkokehityksen suuntaamiseen liittyviä kysymyksiä. Kehitysryhmä jalostaa mahdollisista kehityssuunnista parhaimmat esitettäväksi.

### **6.1 Pohjatutkimustietomoduuli**

Kehitysryhmässä alustavasti pohdittiin yrityksen eri lähteistä tilaamien ja saamien tietojen tallennusprosesseja sekä sitä, kuinka tietoja voitaisiin hyödyntää paikkatietojärjestelmässä, jolloin tietoja voitaisiin tarkastella maantieteellisessä asiayhteydessä. Projekteilla usein käytetään pohjatutkimustietoja, joiden tiedot ovat inframodel- tai tekla-muodossa, ja niiden tiedot ovat usein projektikohtaisia, joten ne olisi luonnollista liittää paikkatietokantaan. Pohjatutkimustiedoista olisi sijainnissaan hyötyä esimerkiksi muille lähialueen suunnittelukohteille, jolloin tarpeellisia ja mahdollisesti hyödyllisiä kairaustietoja ei tarvitsisi etsiä paikallisen palvelimen nykyisestä tiedostorakenteesta.

Yhtenä toteutusvaihtoehtona olisi esittää jokaisen kairauksen tiedot suoraan kartalla, jolloin niitä voitaisiin tarkastella karttanäkymässä suhteessa projektien sijainteihin. Kyseinen toteutustapa ei ole niin käytännöllinen, sillä pohjatutkimustiedostoja käytetään suunnitteluovelluksissa ja ne eivät ole yhteydessä

paikkatietojärjestelmään. Eri kairaustiedostoformaattit ja koordinaatistot aiheuttaisivat ongelmia, sillä paikkatietojärjestelmän tulisi ymmärtää eri formaatteja sekä kääntää kairauspisteiden koordinaatit WGS84-koordinaateiksi.

Parempi ja käytännöllisempi vaihtoehto olisi integroida koko kairaustiedostojen tallennusprosessi osaksi paikkatietojärjestelmää. Silloin paikkatietosovelluksen avulla prosessi olisi mahdollista vakioida eikä käyttäjien välisiä eroja tallennustavoissa esiintyisi. CRM-järjestelmässä on tiedostojenhallintaominaisuuksia, joten tallentamalla kaikki kairaustiedostot sinne olisi ne heti yhdistetty sille kuuluvalla projektille. Käytössä olevat suunnitteluovellukset kuitenkin vaativat, että tiedostot ovat saavutettavissa paikallisella palvelimella eikä tiedostoja ole järkevää tallentaa kahteen sijaintiin. Kehitysryhmä lähti alustavasti tutkimaan nykyisen tallennustavan kehittämistä siihen suuntaan, että paikkatietojärjestelmä yhdistäisi tiedostot kirjauksille, mutta tiedostojen tallennussijainti pysyisi paikallisella palvelimella.

Paikkatietosovellukseen lisättäisiin pohjatutkimustiedon tallennusmoduuli, johon käyttäjä syöttäisi pohjatutkimustiedoston ja projektin, jolle tiedosto kuuluu. Samalla lisätietona tiedostolle lisättäisiin tieto siitä, onko kyseessä uutta vai vanhaa pohjatutkimustietoa. Uudella tiedolla tarkoitetaan yrityksen projektia varten tilaamia maaperätutkimuksia ja vanhalla tiedolla tarkoitetaan tutkimuksia, jotka on saatu muista lähteistä ja toteutettu ennen projektin aloitusta.

Tiedoston syöttämisen jälkeen paikkatietosovellus siirtäisi tiedoston paikalliselle palvelimelle järjestelmän vakioimaan ja ylläpitämään kansiorakenteeseen. Rakenteesta löytyisi projektinumeroidut kansiot, joissa jokaisella olisi alikansiot nimillä *uudet* ja *vanhat*, joihin tallennettava tiedosto siirrettäisiin käyttäjän valinnan perusteella. Käyttäjät voisivat edelleen tallentaa pohjatutkimustiedostot nykyisellä tuntemallaan tavalla, mutta käyttäjien tulisi noudattaa uutta kansiorakennetta. Paikkatietojärjestelmä tekisi pyydettyä tarkastuksia kansiorakenteen eheyden ylläpitämiseksi ja nostaisi esille kansiorakenteen virheet.

Paikalliselle palvelimelle tallentamisen jälkeen pohjatutkimustiedosto on yhdistettävä kirjaukselle. Paikkatietojärjestelmän tallentamistapahtuman yhteydessä voidaan paikkatietokannan kohdekirjauksen dokumentille lisätä merkinnät pohjatutkimustiedon tilasta ja tyypistä. Dokumentille lisättäisiin objekti nimeltä *kairaus* ja tietona sille liitettäisiin tyyppi totuusarvona (engl. *Boolean*), jonka tosiarvo kuvaisi uutta tietoa ja epätosiarvo vanhaa tietoa. Toisaalta tyyppin tieto voisi puuttua objektista kokonaan, jolloin voitaisiin tulkita, että pohjatutkimustietoja ei ole ollenkaan liitetty tiedostolle. Kuitenkin selkeämpää olisi kirjata dokumenteille toinen objekti nimeltä tiedosto ja lisätä siihen totuusarvo, jonka tosiarvo tarkoittaisi, että kirjaukselle on liitetty yksi tai useampi kairaustiedosto ja epätosiarvo tarkoittaisi, että kirjaukselle ei ole ollenkaan liitetty kairaustietoja.

Tyypikuvauksen ja tiedoston olemassaolon kirjaamisen lisäksi dokumenteille lisättäisiin objekti nimellä polku, jolle lisätään tekstimuotoisesti tiedoston tallennussijainti paikallisella palvelimella. Silloin paikkatietosovelluksen karttanäkymän tietokunassa esitettäisiin linkki kyseisen kohteen kairaustiedoston sijaintikansioon. Näin käyttäjät voivat kartalta löytää kairaustiedostot ja ottaa niiden polut heti suunnittelusovelluksen käyttöön, kopioida tiedostot sijainnistaan muualle tai toimittaa ne eteenpäin muille toimijoille.

Kairaustiedon tallentamistavan muutokset ja toimintatavan integroiminen osaksi paikkatietojärjestelmää sai osakseen suhteellisen positiivista kannatusta, mutta kannatus jäi vaisuksi, kun toteuttamisen resurssit otettiin huomioon. Kairaustiedostojen tallentamistavan muutoksiin ei toistaiseksi tartuta eikä niitä todennäköisesti esitetä toteutettavaksi lähitulevaisuudessa.

## 6.2 Karttatoiminnot

Paikkatietosovelluksen karttanäkymään on tarkoitus luoda lisätoiminnallisuuksia, joilla edistettäisiin paikkatiedon analysointimahdollisuuksia. CRM-järjestelmän tiedoista on tarkoitus luoda lisää suodatuskriteerejä, kuten talouslukuja, osallistujat ja asiakkaan toimiala. Kriteerejä voisi ketjuttaa rajattomasti, eli aktivoida kartalle monta suodatinkriteeriä samanaikaisesti.

Paikkatietosovelluksen karttanäkymään on tarkoitus luoda lisätoiminnallisuuksia, joilla edistettäisiin paikkatiedon analysointimahdollisuuksia. CRM-järjestelmästä tuotaisiin lisää tietoja, kuten talouslukuja ja keskeisiä kannattavuusindikaattoreita ja valjastettaisiin niitä suodatinkriteereiksi. Suodattimia kehitetään niin, että kriteerejä voisi ketjuttaa rajattomasti, eli kartalla voisi olla samanaikaisesti aktiivisena monta suodatuskriteeriä. Suodattimien kaveriksi karttanäkymään lisättäisiin piirtotyökaluja, joilla käyttäjä voi piirtää kartalle alueen, jonka sisä- tai ulkopuolelle suodatinkriteerit olisi mahdollista asettaa aktiivisiksi.

Kartalla esitettävän tiedon määrän kasvaessa valtavaksi voidaan harkita karttamerkkien klusteroinnin käyttöönottoa. Klusteroinnilla lähemmäksi olevat karttamerkit yhdistetään yhdeksi suuremmaksi merkiksi, jonka numero osoittaa klusterin sisällä olevien karttamerkkien määrän. Klusteroinnilla kartan luettavuutta saadaan parannettua, mutta toistaiseksi klusteroinnille ei ole ollut tarvetta. Karttamerkkien klusterointia on kuitenkin kehitysvaiheessa kokeiltu ja sen koodi on kirjoitettu jo valmiiksi, joten se voidaan tarvittaessa ottaa käyttöön ja mukauttaa senhetkisiin tarpeisiin.

Karttatoimintojen jatkokehittäminen on keskeistä paikkatietojärjestelmän kannalta, sillä karttatoiminnot edesauttavat paikkatiedon analysointia. Karttatoimintojen toteuttamisidea sai käyttäjäkokeilun perusteella vaihtelevasti kannatusta osakseen. Loppukäyttäjät eivät myöskään olleet vakuuttuneita toimintojen kehittämisen tarpeellisuudesta, kun huomioon otettiin toimintojen toteuttamiseen kuuluvien resurssien arvio. Karttatoiminnot ja niillä tehtävä paikkatiedon analysointi ovat kuitenkin tärkeitä, ellei tärkeimpiä, osia paikkatietojärjestelmässä, joten ideaa ja toteutustapaa hiotaan ja niitä esitetään toteutettavaksi paikkatietojärjestelmän käyttöönoton jälkeen.

### 6.3 Verkkokartta

Yksi ensimmäisistä paikkatietojärjestelmän käyttökohdeideoista oli, että paikkatietokannasta vietäisiin kohteiden tietoja yrityksen verkkosivustolla julkaistavalle kartalle, joka toimisi ikään kuin yrityksen referenssilistana. Verkkokarttaesitys

palvelisi potentiaalisia asiakkaita, jotka voisivat tarkastella kartalta yrityksen kohteita heitä kiinnostavalta alueelta. Paikkatietojärjestelmän avulla verkkokartalla julkaistava tieto olisi aina ajantasaista, ja uusia projekteja ei tarvitsisi erikseen lisätä kartalle. Verkkokartan ilme saadaan myös räätälöityä sopimaan yrityksen uuteen visuaaliseen ilmeeseen sopivaksi.

Paikkatietokanta ja CRM-järjestelmä sisältävät arkaluontoistakin tietoa, joten verkkokartalle poimittaisiin kirjauksilta ainoastaan projektin nimi, toteutuksen aikaväli ja kuvaus kohteesta. Jokaista projektia ei ole tarkoituksenmukaista julkaista verkkokartalla, ja siksi kirjauksia voidaan piilottaa. Paikkatietokannan dokumenteille on jo nyt osoitettu tosiarvo, jonka perusteella piilottaminen tehdään. Paikkatietojärjestelmän automaattisessa tiedon käsittelyssä kirjat tul- laan oletusarvoisesti piilottamaan, ja käyttäjä voi halutessaan erikseen julkaista kohteen tai valita kohteen julkaistavaksi sijainnin syötön yhteydessä. Oletusarvoisella piilottamisella minimoidaan vahingolliset tiedon julkaisun riskit tilan- teissa, joissa yrityksellä ei ole oikeutta tiedottaa hankkeen aloituksesta tai hanke on muulla tavoin salattu.

Ensiarvoisen tärkeää on, että paikkatietojärjestelmän sensitiiviset tiedot eivät vuoda organisaation ulkopuolelle tietoturvahyökkäyksessä, ja internetissä tietoja julkaistaessa tietoturvariskit muutenkin kasvavat huomattavasti (3). Siksi verk- kokarttajulkaisussa ei tietoja haeta suoraan varsinaisista tietokannoista, vaan paikkatietokannan dokumentit kloonataan esimerkiksi pilvipohjaiselle MongoDB Atlas -tietokanta-alustalle. Kloonatuilta dokumenteilta karsittaisiin ylimääräiset tiedot, jotka ei ole tarkoitettu julkaistaviksi ja Atlas-tietokanta päivitetäisiin paikkatietojärjestelmästä viikon välein ajantasaisuuden vahvistamiseksi. Paikka- tietojärjestelmän paikallista palvelinta ei missään tapauksessa tule perusteetto- masti liittää internetiin, vaikka palvelimen palomuuuri on ajan tasalla ja tietotur- vallisuutta monitoroidaan aktiivisesti.

## 7 Yhteenveto

Kehitysprojektissa toteutettiin paikkatietojärjestelmä aivan ruohonjuuritason tutkimusvaiheesta käyttöönottovalmiiksi kokonaisuudeksi. Paikkatietojärjestelmän paikkatietosovelluksen avulla ulkoisen palveluntarjoajan asiakkaiden- ja projektienhallintajärjestelmän (CRM-järjestelmä) aineisto yhdistetään uuden paikkatietokannan kanssa. Sovellus muokkaa tietoja analysoitavaan muotoon halutulle tarkkuustasolle ja liittää paikkatiedon CRM-järjestelmän kirjauksille automaattisilla toimintamalleilla, joilla vähennetään järjestelmän ylläpidon tarvetta verrattuna tavanomaisen karttaesityksen luontiin. Yrityksen sisäisellä sovelluskehityksellä saavutettiin kehitysprojektin alussa paikkatietojärjestelmälle asetetut tavoitteet ja lisättiin onnistuneesti nykyisen asiakkaiden- ja projektienhallintajärjestelmän sisällön arvoa, vaikka toteutuksen laajuutta täytyi tiukasti rajata insinööriyön puitteisiin.

Paikkatiedon avulla CRM-järjestelmän tietoja on mahdollista tarkastella maantieteellisen sijainnin asiayhteydessä, mutta sijaintitiedosta ei ole sen suurempaa hyötyä ilman kaikkea sitä tukevaa tietoa. CRM-järjestelmän tietomäärä on valtava, mutta tietosisällöstä ja sen eheydestä muodostuu haaste kehitysprojektin päätavoitteille, jos loppukäyttäjää ei saada aktivoitua CRM-järjestelmän täydentämiseen. Sen vuoksi projektilla keskityttiin paikkatietoteknisten toimintamallien toteuttamisen lisäksi CRM-järjestelmän muun sisällön kehitykseen sekä vakiointiin, jotta aineisto palvelisi mahdollisimman hyvin paikkatietojärjestelmän tarpeita. Käyttöönottovalmiin järjestelmän kehityksen menestys ei ollut tähtiin kirjoitettu, ja projekti kohtasi odottamattomiakin haasteita, mutta niistä suoriututtiin menestyksekkäästi. Kehitystyön tavoitteiden saavuttamisen lisäksi projektin aikajänteellä oivallettiin uutta, kirkastettiin koko paikkatietojärjestelmän ydintoimintaideaa ja sen mahdollisuuksia ja lopulta kokonaisuus nähtiin tavalla, jota ei olisi projektin alussa osattu arvioida. Paikkatietojärjestelmä kannustaakin CRM-järjestelmän tietojen sääntilliseen ylläpitoon, sillä kirjauksille lisättävä arvo realisoituu käyttäjälle saman tien ajantasaisen karttaesityksen muodossa. Kehitysprojektin lopulla paikkatietojärjestelmä nähdään kokonaisuutena, jossa

paikkatietosovellus ja CRM-järjestelmä toimivat ikään kuin symbioosissa keskenään, tukien toisiaan.

Paikkatietojärjestelmän kehitystyö jatkuu käyttööntöivaiheella, jota lähdetään välittömästi valmistelemaan käyttööntöosuunnitelman mukaisesti (liite 1). Paikkatietosovelluksen kehityksen jatkumo näyttää lupaavalta, sillä yrityksen yksi tärkeimmistä arvoista on jatkuva toiminnankehittäminen. Uusilla paikkatietojärjestelmän mahdollistamilla toimintamalleilla edistetään CRM-järjestelmän todellisen potentiaalin avaamista ja jatkossa järjestelmään lisätään paikkatiedon analysointimahdollisuuksia.

## Lähteet

- 1 In-House Development vs Outsourcing: Which is the best option? 2021. Verkkoaineisto. Dreamcode. <<https://www.dreamcode.io/blog/in-house-development-vs-outsourcing>>. Päivitetty 8.2.2021. Luettu 23.2.2022.
- 2 Desktop or Web Application: What to Develop? 2021. Verkkoaineisto. Exoft LCC. <<https://exoft.net/desktop-or-web-application-what-to-develop/>>. Päivitetty 21.12.2021. Luettu 17.3.2022.
- 3 What is web application security? 2022. Verkkoaineisto. Cloudflare Inc. <<https://www.cloudflare.com/learning/security/what-is-web-application-security/>>. Luettu 7.4.2022.
- 4 Introduction to HTML. 2021. Verkkoaineisto. W3Schools. <[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)>. Luettu 15.11.2021.
- 5 CSS: Cascading Style Sheets. 2022. Verkkoaineisto. Mozilla. <<https://developer.mozilla.org/en-US/docs/Web/CSS>>. Päivitetty 21.1.2022. Luettu 13.3.2022.
- 6 JavaScript. 2021. Verkkoaineisto. Mozilla. <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Päivitetty 29.7.2021. Luettu 13.3.2022.
- 7 High-level programming language. 2022. Verkkoaineisto. Mozilla. <[https://developer.mozilla.org/en-US/docs/Glossary/High-level\\_programming\\_language](https://developer.mozilla.org/en-US/docs/Glossary/High-level_programming_language)>. Päivitetty 18.2.2022. Luettu 13.3.2022.
- 8 Carbonnelle, Pierre. 2022. PopularitY of Programming Language Index. Verkkoaineisto. PYPL. <<https://pypl.github.io/PYPL.html>>. Luettu 13.3.2022.
- 9 About JavaScript. 2021. Verkkoaineisto. Mozilla. <[https://developer.mozilla.org/en-US/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript)>. Päivitetty 26.7.2022. Luettu 23.2.2022.
- 10 Herron, David. 2018. Node.js Web Development: In Server-Side Development with Node 10 Made Easy, 4<sup>th</sup> Edition. E-kirja. Packt Publishing.
- 11 Introduction to Node.js. 2021. Verkkoaineisto. OpenJS Foundation. <<https://nodejs.dev/learn>>. Luettu 16.11.2021.

- 12 Process Model. 2021. Verkkoaineisto. OpenJS Foundation and Electron contributors. <<https://www.electronjs.org/docs/latest/tutorial/process-model>>. Luettu 17.11.2021.
- 13 Building web apps in WebView. 2021. Verkkoaineisto. Google Developers. <<https://developer.android.com/guide/webapps/webview>>. Päivitetty 10.12.2021. Luettu 7.4.2022.
- 14 Promise – JavaScript. 2022. Verkkoaineisto. Mozilla. <[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise)>. Päivitetty 18.2.2022. Luettu 23.2.2021.
- 15 Gupta, Lokesh. 2021. Comparing SOAP vs REST APIs. Verkkoaineisto. <<https://restfulapi.net/soap-vs-rest-apis/>>. Päivitetty 27.9.2021. Luettu 17.3.2022.
- 16 SQL Introduction. 2022. Verkkoaineisto. W3Schools. <[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)>. Luettu 14.3.2022.
- 17 What is NoSQL? NoSQL Databases Explained. 2021. Verkkoaineisto. MongoDB. <<https://www.mongodb.com/nosql-explained>>. Luettu 14.3.2022.
- 18 Share, download, or print your map. 2021. Verkkoaineisto. Google. <<https://support.google.com/mymaps/answer/3109452>>. Luettu 31.10.2021.
- 19 Import and Export Data. 2021. Verkkoaineisto. MongoDB. <<https://docs.mongodb.com/compass/current/import-export>>. Luettu 23.2.2022.
- 20 Customize and package your Electron app with OS-specific bundles (.app, .exe, etc.) via JS or CLI. 2022. Verkkoaineisto. Github Contributors. <<https://github.com/electron/electron-packager>>. Päivitetty 4.3.2022. Luettu 7.4.2022.
- 21 Geocoding API Overview. 2022. Verkkoaineisto. Google Developers. <<https://developers.google.com/maps/documentation/geocoding/overview>>. Päivitetty 16.3.2022. Luettu 17.3.2022.
- 22 Geocoding API. 2022. Verkkoaineisto. Google Developers. <<https://developers.google.com/maps/documentation/geocoding/usage-and-billing>>. Päivitetty 6.4.2022. Luettu 7.4.2022.
- 23 What is a Key Performance Indicator (KPI)? 2022. Verkkoaineisto. KPI.org. <<https://kpi.org/KPI-Basics>>. Luettu 7.4.2022.

- 24 Ojanpää, Helena; Näsänen, Risto; Kojo, Ilpo. 2002. Eye Movements in the Visual Search of Word Lists. *Vision Research* Volume 42, Issue 12, 12.4.2002, s. 1499–1512.



## **Paikkatietojärjestelmä**

### **Liite 1. käyttöönottosuunnitelma**

7.5.2022



## Sisällysluettelo

1	Yleistä .....	1
1.1	Kokonaisuus lyhyesti .....	1
2	Riskienhallinta.....	1
2.1	CRM-järjestelmän ohjelmointirajapinta .....	1
2.2	Kehitystiimin järjestäytyminen .....	1
2.3	Tiedon turmeltuminen .....	2
2.4	Käyttöönoton laajuus .....	2
2.5	Jatkokehitys .....	2
3	Välittömät toimenpiteet.....	2
4	Käyttäjäkokemusta parantavat toimenpiteet .....	3

## 1 Yleistä

### 1.1 Kokonaisuus lyhyesti

Käyttöön otettava paikkatietojärjestelmä sisältää paikkatietokannan ja sen ylläpitoa edistävän sekä paikkatiedon tarkastelua varten kehitetyn paikkatietosovelluksen. Paikkatietojärjestelmällä täydennetään CRM-järjestelmän tietoja ja järjestelmät ovat yhdistettävissä toisiinsa. Paikkatietojärjestelmässä CRM-järjestelmän tiedot esitetään maantieteellisessä asiayhteydessä ja sovelluksella on mahdollista tehdä paikkatietoanalyyskejä.

## 2 Riskienhallinta

Käyttöönottoa ennen on tunnistettu riskejä, jotka voivat vaikuttaa käyttöönottoon ja sen jälkeisiin tehtäviin.

### 2.1 CRM-järjestelmän ohjelmointirajapinta

Kyseisen ohjelmointirajapinnan määritykset, kuten päätepisteiden osoitteet sekä pyyntöjen muotoilut voivat aiheuttaa odottamattomia muutostarpeita paikkatietosovelluksen koodille.

Riskien ehkäisymenpiteet:

Seurataan ulkoisen palveluntarjoajan tiedotteita ohjelmointirajapinnan muutoksista ja selvitetään tarvittaessa tulevien muutosten aikataulutusta. Näin yllättäviä resurssitarpeita voidaan ennakoida ja kohdentaa niitä paikkatietosovelluksen osioille ilman ylimääräistä häiriötä. Resurssitarpeita arvioidaan ennalta, jotta kehitystiimi osaisi makrotasolla varautua muutosteh-teisiin.

### 2.2 Kehitystiimin järjestäytyminen

Paikkatietojärjestelmän kehitystiimin ja sen osaamisen sisäinen järjestäytyminen yrityksessä voi muuttua, jolloin siitä syntyy riskitekijä järjestelmän käyttöönotolle ja sitä seuraavalle ylläpidolle sekä jatkokehitykselle.

Riskien ehkäisymenpiteet:

Pyritään pitämään tiimi kasassa ja uusien työntekijöiden kohdalla selvitetään kiinnostusta osallistua paikkatietojärjestelmän kehitystiimiin.

Kehitystiimi pyritään pitämään kasassa ja uusien työntekijöiden kohdalla kartoitetaan kiinnostusta osallistua paikkatietojärjestelmän kehitykseen. Uudet jäsenet perehdytetään asianmukaisesti ja yrityksen tahtotilana on osoittaa resursseja projektille pitäen sen jatkuvan kehitystyön mielekkäänä.

## 2.3 Tiedon turmeltuminen

Sovelluksen koodi tai sen osa voi turmeltua esimerkiksi kovalevyn osion menetyksen tai laitevian seurauksena. Tiedon turmeltumiseen on myös rinnastettavissa tilanteet, joissa tietokantojen eheys tai alkuperäinen tietosisältö muuttuisi odottamattomasti esimerkiksi bugin seurauksena.

Riskien ehkäisytöimenpiteet:

Varmuuskopioidaan paikkatietosovellus ja -tietokanta säännöllisesti.

## 2.4 Käyttöönoton laajuus

Käyttöönottovaiheen tehtävät on pidettävä vaiheelle varattujen resurssien rajoissa. Mikäli arvio ylittyy, niin käyttöönottovaihe voi venyä odottamattomasti ja resursseja ei välttämättä ole käytettävissä tai vapautettavissa, joka pidentää käyttöönoton aikataulua.

Riskin ehkäisytöimenpiteet:

Tarkastellaan käyttöönottosuunnitelmassa esitettyjen arvioiden paikkansapitävyyttä ja tarkennetaan arvioita tarvittaessa. Ennen kaikkea on varattava keskeytyksetöntä aikaa, jotta käyttöönotto saadaan mahdollisimman nopeasti hoidettua.

## 2.5 Jatkokehitys

Organisaatiossa voi tapahtua odottamattomia muutoksia henkilöstön vaihdosten myötä tai johdon intressit voivat muuttua. Negatiivisesti päätöksen tekoon ja jatkokehitykseen voi vaikuttaa esimerkiksi odottamattomat käyttöönoton haasteet tai aikataulun ja resurssiarvioiden paikkansapitämättömyys.

Kehitysvaiheen päättymisen lopulla paikkatietojärjestelmän kehityksen jatkuvuuden näkymät ovat hyvät, sillä organisaatiossa on onnistuneesti aktivoitu henkilöstöä CRM-järjestelmän käyttöön. Jatkokehityksen suuntauksia on hyvä valmistella huolella esitettäväksi, jotta päätöksenteko pysyy mutkattomana.

Riskin ehkäisytöimenpiteet:

Aktivoidaan työntekijöitä paikkatietojärjestelmän käyttöön ja pidetään organisaation johdolle selkeitä esityksiä jatkokehityksen suuntautumisesta, jotta intressit järjestelmän kehittämiseen säilyisivät.

## 3 Välittömät toimenpiteet

Käyttöönottovaiheessa on tehtävälliställä välittömiä toimenpiteitä, jotta paikkatietojärjestelmä saadaan toimintavalmiuteen käyttöönottoympäristössä.

toimenpide	kuvaus
CRM-järjestelmän tietokantamerkintöjen tallennussijainnin muuttaminen	Paikkatietosovellus tuottaa tietokantamerkintöjä suoraan CRM-järjestelmään ja merkinnän tallennussijainnissa voi osittain olla jo tallennettua tietoa. Ennen käyttöönottoa merkintöjen tallennussijainti on tyhjennettävä ja siellä nyt sijaitsevia tietoja tallennetaan muualle. Toisena vaihtoehtona on siirtää tietokantamerkinnot toiseen tallennussijaintiin.
paikkatietosovelluksen pyynnöt CRM-järjestelmän ohjelmointirajapintaan	Paikkatietosovellus lähettää pyyntöjä CRM-järjestelmän rajapintaan ja kaikki pyynnöt on systemaattisesti tarkastettava ja päivitettävä vastaamaan ohjelmointirajapinnan viimeisimmän version mukaisia määrittäyksiä.
paikkatietojärjestelmän toiminnan varmistaminen	Sovelluksen toiminta todellisessa CRM-ympäristössä on tarkastettava tietokantatoimintojen tarkastusluettelon mukaisesti (liite 2).
Google Maps asiakkuudenhallinta	Google Maps Platform asiakkuus siirretään yrityksen hallintaan, jolloin sovelluksen käyttämä API-avain on vaihdettava uuteen.
tietokannan siirto	Paikkatietokanta siirretään paikalliselle palvelimelle.
tunnistelistojen vakiointi	Käyttöön otettavista avainsana- ja kategorialistoista laaditaan lopulliset versiot ja otetaan ne käyttöön CRM-järjestelmässä.
paikkatietosovelluksen paketointi	Paikkatietosovellus paketoitetaan asennustiedostoksi electron packagerilla.

#### 4 Käyttäjäkokeista parantavat toimenpiteet

Kehitysvaiheen loppupuolella on tunnistettu parannuskohteita ja muutamia virheitä, jotka liittyvät paikkatietosovelluksen yleiseen käyttökokeeseen. Kyseisiä toimenpiteitä ei ole välttämätöntä toteuttaa käyttöönottovaiheessa, mutta niiden rekisteröimiseksi ne ovat kirjattuna käyttöönottosuunnitelmaan.

Käyttäjäkokeista parantavat toimenpiteet ovat jaoteltu taulukkoon sovellusosuuksittain seuraavasti:

<b>kohde ja toimenpide</b>	<b>kuvaus</b>
<b>Tietokantamoduuli</b> projektijanan toiminta	Projektijanalla uusi kirjaus ei ilmesty näkyviin, jos janaa siirretään näyttämään vain yhtä kirjausta, jonka jälkeen jana palautetaan näyttämään kahta kirjausta. Kadonnut kirjaus palautuu näkymään, kun janaa siirretään esittämään useampaa kuin kahta kirjausta.  Projektijanan toiminta systemaattisesti liikuttaa erästä toimintonäppäintä näkymässä.  Projektijanan välin projektinumerot näkyvät ainoastaan janan päätepisteitä siirrettäessä. Muutetaan numerot aina esitettäväksi.
<b>Tietokantamoduuli</b> kirjausluettelon ja kartan asettelu	Kirjausluettelo siirretään näkymän ylälaidasta näkymän vasempaan reunaan ja kartta alalaidasta oikeaan reunaan. Näin näkymän tila saadaan tehokkaampaan käyttöön, kun luettelo ja kartta ovat pystysuorassa.
<b>Tietokantamoduuli</b> ohjeistuksen saavutettavuus	Asetetaan tietokantamoduulin käyttöohjeet paremmin käyttäjien saavutettaviin.
<b>Karttanäkymä</b> hakutoiminnon sijoittelu	Hakutoiminnon sijainti ei ole paras mahdollinen, joten siirretään se oikeasta ylälaidasta keskeisempään sijaintiin, jotta se on paremmin käyttäjien tavoitettavissa.
<b>Karttanäkymä</b> tietoikkunan tiedot	Lisätään CRM-järjestelmästä tietoja näkyviin karttanäkymän tietoikkunaan.  Harkitaan tietoikkunassa avainsanojen pienentämistä.
<b>Yleisesti</b> paikkatietosovelluksessa käytetyn kielen muutos	Paikkatietosovelluksessa on käytetty sekaisin englantia ja suomea, joten vaihdetaan koko sovellus suomen kielelle.
<b>Yleisesti</b> Koko sovelluksen värimaailman muutos	Yrityksen graafinen ilme on muuttunut sovelluskehityksen aikana ja paikkatietosovellus voitaisiin mukauttaa vastaamaan sitä.

## CRM-järjestelmän tietokannan kehitysympäristön aineisto ja sille tehtävät tarkastukset

✔ toiminta ok
 ✘ virheellinen toiminta
 ? toimii osittain

Projektinnumero	Kuvauksen sisältö	Arvo	Tarkastuksen tarkoitus	Tapahtuma	Tietokantamerkintä	Haasteet	Toiminta
1005	todellinen osoite	tyynylaavantie 5, helsinki	Varmistetaan, että geokoodauksen ohjelmointirajapinta tulkitsee todellisen osoitteen oikein.	Syötetty arvo geokoodataan Google Mapsin geokoodaus ohjelmointirajapinnalla. Rajapinta palauttaa osoitteen WGS84-muotoisina koordinaatteina, jotka tallennetaan paikkatietokannan dokumentille.	m3	Google Mapsin geokoodaus-rajapinta tulkitsee osoitteet tarkasti. Jos käyttäjä ei syötä tarpeeksi yksityistä osoitetietoa esimerkiksi muotoilulla osoite ja kunta, niin useamman saman nimisen osoitteen joukosta valikoituu käyttäjän sijaintiin nähden lähin osoite.	✔
1006	tuntematon osoite	testiosoite 15	Varmistetaan, että geokoodauksen ohjelmointirajapinta tulkitsee tuntemattomaksi osoitteen, jota ei ole olemassa.	Syötetty arvo geokoodataan Google Mapsin geokoodaus ohjelmointirajapinnalla. Rajapinta palauttaa geokoodauksen statuksena muun kuin OK, jolloin geokoodaus epäonnistuu.	m1	Google Mapsin geokoodaus-rajapinta tulkitsee hyvin vapaasti osoitteet. Osoitteen arvon ollessa jonkin tarkoitettun osoitteen kaltainen, niin sijainti voi siirtyä väärään paikkaan tai väärään maahan, ellei toimintaa rajoiteta.	✔
1007	normaalit koordinaatit	60,25	Tarkastetaan, että koordinaatit tallentuvat oikein paikkatietokantaan ja siitä jätetään merkintä CRM-järjestelmän tietokantaan.	Syötetty arvo muotoillaan ja tallennetaan sellaisenaan paikkatietokantaan.	m3	Käyttäjällä voi vahingossa tapahtua näppäilyvirhe.	✔
1008	muotoillut koordinaatit	60.999 , 25.999	Tarkastetaan, että koordinaatit rekisteröityvät oikein paikkatietokantaan, vaikka käyttäjä kirjoittaisi koordinaatit eri muotoilulla.	Syötetty arvo muotoillaan ja tallennetaan sellaisenaan paikkatietokantaan.	m3	Jokaisen muotoilun toiminta tulisi erikseen tarkastaa.	✔
1009	tuntemattomat koordinaatit	6800000,25000000	Tarkastetaan, ettei sovellus hyväksy toisessa koordinaattijärjestelmässä ilmoitettuja koordinaatteja.	Merkkijonoa yritetään geokoodata, mutta geokoodaus epäonnistuu, sillä koordinaatteja ei tunneta.	m1	Googlen geokoodaus saattaa tulkita syötetyn arvon osoitteeksi.	✔
1010	jo käsitelty kirjaus ja sen merkintä	m1, m2 tai m3	Kun kirjaus on kerran luettu paikkatietokantaan sitä ei ole syytä lukea uudelleen, ellei siinä ole virhettä. Käyttäjää ohjataan tietokantamerkinnällä toimimaan tilanteessa, jossa rekisteröity tieto on virheellistä.	Kirjaus ohitetaan.	ei käsitellä, säilytetään oleva merkintä		✔
1011	tyhjä	tyhjä	Tallennetaan uuden kirjauksen tiedot paikkatietokannan dokumentille muotoiltuna.	Kirjaus tallennetaan paikkatietokantaan ja merkitään se luetuksi.	m1		✔

## CRM-järjestelmän tietokannan tapahtumaluettelo

Toiminto	Tapahtuma	Haasteet	Toiminta
<b>Kirjausten hallinta</b>			
uusi kirjaus	Kirjaus siirretään tilasta riippuen käsiteltäväksi.		
tilan muutos tilaukseksi	Kirjaus siirretään käsiteltäväksi.		
tilan palautus tarjoukseksi	Kirjaus poistetaan paikkatietokannasta ja nollataan CRM-järjestelmän tietokannassa.		
kirjauksen poisto	Kirjaus poistetaan paikkatietokannasta ja nollataan CRM-järjestelmän tietokannassa.		
<b>Kirjauksen ominaisuuksien muutokset</b>			
nimen muutos	Päivitetään muutettu nimi paikkatietokannan dokumentille.		
tilan muutos	Päivitetään uusi tila paikkatietokannan dokumentille.		
oletusasiakkaan muutos	Päivitetään asiakkaan nimi paikkatietokannan dokumentille.		
<b>Avainsanojen hallinta</b>			
tunnisteen lisäys avainsanalistaan	Päivitetään avainsanan aktiivisuus karttanäkymän tietoiikkunaan.		
tunnisteen poisto avainsanalistalta	Päivitetään avainsanan aktiivisuus karttanäkymän tietoiikkunaan.		
tunnisteen käytöstä poisto	Avainsanalista päivittyy karttanäkymän tietoiikkunaan.		

## Paikkatietokannan dokumentin muotoilu

Tietosisältö	Nimike	Tietomuoto	Tiedon määrittely	Käyttötarkoitus	Mahdolliset haasteet
yksilöivä tunniste (UUID)	_id	merkkijono	Kirjaus perii CRM-järjestelmän tietokannan kirjaukselta UUID:n, eli se kopioidaan paikkatietokannan dokumentille.	Toimii rajapintojen välisen tiedonsiirron yksilöivänä tunnisteena. Voidaan esimerkiksi poimia kummastakin tietokannasta paikkatietosovellukseen esitettäväksi tietoa ja voidaan varmistua, että tiedot ovat oikealta kirjaukselta.	Ei tiedossa
yrittäjän nimi	company	merkkijono	Vakiona yrityksen nimi, joka on toistaiseksi staattinen tieto, eikä sitä tarvitse kopioida tietokannasta.	Samalla kartalla voidaan esittää useamman kuin yhden yrityksen projekteja.	Tietokantojen yhdistäminen voi osoittautua haasteelliseksi. Ulkoisen tietokannan yhdistäminen voisi olla esimerkiksi istuntokohtaista, eli yhteys tulisi lisätä erikseen karttanäkymässä valitsemalla lisättävä yritys. Lisättävän yrityksen tiedot voisi olla eri paikkatietokannassa.
projektin numero	job_number	kokonaisluku	Kirjaus perii CRM-järjestelmän kirjaukselta projektinumeron, eli se kopioidaan paikkatietokannan dokumentille.	Projektinumero on esisijainen tunniste loppukäyttäjille. Sillä voidaan hakea karttaliittymässä tietyn projektin sijaintia ja esittää projektinumero tunnistetietona.	Ei tiedossa
projektin nimi	job_name	merkkijono	Kirjaus perii CRM-järjestelmän kirjaukselta projektin nimen, eli se kopioidaan paikkatietokannan dokumentille.	Projektin nimi on toissijainen tunniste loppukäyttäjille. Projektin nimi esitetään usein tarkentavana tunnisteena projektinumeron yhteydessä.	Ei tiedossa
projektin tila	job_status	merkkijono	CRM-järjestelmästä poimitaan kirjauksen isClosed-tilan boolean-arvo ja se muotoillaan välitietokannan dokumentille luettavaksi arvoksi. Kirjauksen epätoisi boolean-arvo tulkitaan avoimeksi kirjaukseksi. Projektin päättyessä boolean-arvo vaihtuu todeksi, jolloin tilaksi tulkitaan dokumentille "suljettu".	Toimii rajaavana tietona karttanäkymän karttamerkkien suodatuksessa.	Ei tiedossa
projektin aloituspäivämäärä	start_date	merkkijono	CRM-järjestelmän kirjaukselta poimitaan tarkka projektin päättymispäivämäärä muodossa pp.kk.vvvv ja se muotoillaan paikkatietokannan dokumentille vuositarkkuuteen.	Toimii rajaavana tietona. Esimerkiksi suodattimessa, kun halutaan esittää projektit, jotka on ollut aktiivisena vuoden 2021 aikana. Päättymispäivämäärän kanssa rajauksen voi tehdä tietyille aikavälille.	Kirjauksen luontipäivämäärä ei aina tarkoita projektilla tehtävän työn aloitusajankohtaa. Luontipäivämäärä vastaa myös myyntitapahtuman aloitusta.
projektin päättymispäivämäärä	end_date	merkkijono	CRM-järjestelmän kirjaukselta poimitaan tarkka kirjauksen luontipäivämäärä muodossa pp.kk.vvvv, joka tulkitaan projektin aloitus päivämääräksi. Päivämäärä muotoillaan paikkatietokannan dokumentille vuositarkkuuteen.	Toimii rajaavana tietona. Esimerkiksi suodattimessa, kun halutaan esittää kartalla projektit, jotka on päättyneet vuonna 2020. Aloituspäivämäärän kanssa rajauksen voi tehdä tietyille aikavälille.	Edes vuositarkkuudella päättymispäivämäärä ei aina vastaa projektin todellista päättymisajankohtaa.
asiakkaan nimi	client	merkkijono	Kirjaus perii CRM-järjestelmän kirjaukselta asiakkaan nimen, eli se kopioidaan välitietokannan dokumentille.	Esitetään karttanäkymän tietoikkunassa tai käytetään rajaavana tietona suodattimissa. Esimerkiksi suodattimessa, kun halutaan esittää kartalla tietyille asiakkaalle tehdyt projektit.	Yksittäisellä projektilla voi olla useampi asiakas, mutta vain yksi oletusasiakas. Esitetäänkö muita asiakkaita vai ainoastaan oletusasiakas, kuten nykyisessä toteutuksessa on tehty? Oletusasiakkaan nimen päivitys voi osoittautua hankalaksi kesken projektin. Kirjauksen luonnin yhteydessä osoitettua oletusasiakasta ei voida poistaa projektilta, vaan oletusasiakas-status voidaan vaihtaa toiselle projektille määritetyille asiakkaalle.

koordinaatit	geometry	taulukko	Paikkatietosovellus tulkitsee sijainnin tai käyttäjä lisää sijainnin sovelluksessa, jolloin määritetyt WGS84-muotoiset koordinaatit tallennetaan paikkatietokannan dokumentille.	Kirjaukselle on liitetty sijaintitieto ja kirjauksen tietoja voidaan esittää maantieteellisessä asiayhteydessä. Projekti voidaan esittää kartalla tai sijainnille ja sen tiedoille voidaan tehdä paikkatiedon analysointia.	Käyttäjän syöttämän osoitteen perusteella tehtävä sijainnin määrittäminen voi olla hieman epätarkka ja poiketa todellisesta sijainnista. Ratkaisuna on paikkatietosovelluksen sijainninsyöttömoduuli, jossa käyttäjä voi siirtää karttamerkin projektin tarkkaan sijaintiin. Osoitteella tehtävässä määrittämisessä on riski, että osoite menee väärään kuntaan tai kaupunkiin. Käyttäjiä voidaan ohjeistaa lisäämään osoitteen perään projektin kunta tai kaupunki.
piilotettu	hidden	boolean	Sovellus tai käyttäjä määrittelee kirjauksen arvon piilotetuksi. Arvo on vakiona tosi, jottei vahingossa arkaluontoista tietoa tulisi julki.	Piilotettu-tiedolla määritellään verkkokartalla julkaistavat projektit. Verkkokartalla esitetään kirjauksen rajoitettu tietosisältö.	Julkisesti internetissä esillä olevassa verkkokartassa tulee kiinnittää erityistä huomiota toteutuksen tietoturvasuuteen.