

Mikko Korhonen

TYÖMAAN ETENEMISEN JA TYÖKONEEN LIIKKEIDEN SEURANTASOVEL- LUKSEN PROTOTYYPIN SUUNNITTELU

TYÖMAAN EDISTYMISEN JA TYÖKONEEN LIIKKEIDEN SEURANTASOVEL- LUKSEN PROTOTYYPIN SUUNNITTELU

Mikko Korhonen
Opinnäytetyö
Kevät 2019
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, älykkäät järjestelmät

Tekijä: Mikko Korhonen

Opinnäytetyön nimi: Työmaan etenemisen ja työkoneen liikkeiden seurantasovelluksen prototyypin suunnittelu

Työn ohjaaja: Kari Jyrkkä

Työn valmistumislukukausi ja -vuosi: Kevät 2019

Sivumäärä: 24 + 3

Työ on aloitettu nykyiselle työnantajalleni, Herman IT Oy:lle, asiakkaalta tulleen yhteydenoton seurauksena. Asiakkaalla oli tarve sovellukselle tai palvelulle (jatkossa sovellus), jolla seurata kentällä olevien työkoneiden liikkumista ja edistymistä tehtävissä. Nykyisin asiakas saa tietoa työn etenemisestä vain soittamalla mm. aliurakoitsijalle ja muille mukana oleville tahoille. Kukaan ei kuitenkaan välttämättä osaa sanoa kovinkaan tarkkaan, missä kohtaa työkone on etenemässä tai onko kone työmaalla. Myöskään mitään järjestelmää, jossa voisi katsoa nykyisiä tai jo sovittuja työmaita, ei ole olemassa.

Karttapohjaisessa järjestelmässä työmaat olisi mahdollista piirtää kartalle, josta niitä voi tarkastella työn ollessa käynnissä. Myös työkoneen etenemistä määritellyllä alueella voi seurata reaaliaikaisesti, jos koneissa olisi paikannuslaite. Tällöin työn valmistumista olisi helpompi arvioida ja valmistautua paremmin uudelle työmaalle siirtymiseen. Opinnäytetyössä on tavoitteena tehdä web-pohjainen prototyyppisovellus, jolla hahmotetaan asiaa ja sen vaatimuksia sovelluksen tekijälle ja asiakkaalle. Tältä pohjalta lähdetään viemään asiaa eteenpäin yhdessä asiakkaan kanssa sovittuun suuntaan.

Prototyyppisovellukseen suunniteltiin näkymäluonnokselle aloitussivu, koneen seurantasivu, ja jokin sivu, jolla on tarkemmin tietoa kohteesta. Näistä oli tarkoitus lähteä kehittämään sovelluksen toimintoja asiakkaan toivomaan suuntaan. Työn tavoitteena oli prototyyppisovelluksen jälkeen saada esittelyssä olleet ja asiakkaan kanssa ensin toteuttavat toiminnot käyttöön. Tavoitteena oli saada myös joitakin muita toimintoja ainakin testikäyttöön, jos ei aivan tuotantokelpoiseen kuntoon.

Prototyyppisovellukseen saatiin toteutettua laitteen sijaintiedon kerääminen ja joidenkin sivujen hahmotelmat. Työn viivästymisen vuoksi muita suunniteltuja toiminallisuuksia ei ehditty toteuttaa. Projektin ehti myös edetä muilla tahoilla ohi suunnitellun esittelyvaiheen. Työssä syntyneet hahmotelmat ja selvitykset ovat kuitenkin käytössä projektin jatko vaiheissa.

Asiasanat: GPS, seuranta, työkoneet, työmaat, Aplicom

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Intelligent Systems

Author: Mikko Korhonen

Title of thesis: Planning of prototype service to follow working area and working machine progress in target

Supervisor: Kari Jyrkkä

Term and year when the thesis was submitted: Spring 2019 Number of pages: 25 + 3

This thesis is made for my current employer, Herman IT Oy. Thesis gets its idea from customer who had difficulties to locate and follow working machines and progress of working areas in field. At this moment if customer wants to get that information, customer must call several contractors and subcontractors and ask situation and location. Often they do not get this information despite those inquiries.

A goal of this thesis was to plan an application or service where customer can markup work areas and working machines. Machines have GPS modules which send location information and shows machine locations in applications or services map. From the application customer could see how and when machine has been moving in working area and mark up planned work start end time. If possible, the application or the service shows end time estimate for that work in addition of planned estimate. Now customer could plan next work areas easier and more accurately. Purpose of this prototype is to be used to help customer to understand benefits of this kind of application or service and help to make decision to order this kind of application or service from us.

A prototype of an application was the goal of this thesis. It did not get ready, and project did proceed other ways over the main goal of this thesis. Collecting of GPS data of machine were only really working function that get ready. All other plans and statements that were made in this thesis can be used later in main project.

Keywords: GPS, tracking, machine, workarea, Aplicom

SISÄLLYS	
1	JOHDANTO6
2	VAATIMUSMÄÄRITTELY7
2.1	Työkoneesta kerättävät tiedot ja tietojen siirto.....7
2.2	Web-sovelluksen vaatimukset.....9
2.2.1	Prototyyppisovellus..... 10
2.2.2	Tietokanta 11
2.2.3	Palvelinalusta..... 12
2.3	Vaatimukset käytettäville karttapohjan työkaluille 12
3	PROTOTYYPPISOVELLUKSEN TOTEUTUSSUUNNITELMA 14
3.1	Aplicom A1 BASIX, datan mittaus ja tiedonsiirto 15
3.2	Aplicom A1 BASIXin konfigurointi 15
3.3	Tietojen vastaanotto ja vastaanotetun datan purkaminen..... 19
3.4	Tietokanta 19
4	TYÖN TILANNE OPINNÄYTE TYÖN PÄÄTÖSVAIHEESSA22
4.1	Taustatoimintojen vaiheet.....22
4.2	Web-sivut.....23
4.3	Oppimiskokemukset.....23
LÄHTEET.....25	

1 JOHDANTO

Työkoneiden ja työmaan seuranta etänä on usein hyvin vaikeaa. On olemassa sovelluksia, joilla edistymistä voidaan seurata, mutta ne ovat usein tietyille isolle organisaatiolle tehtyjä ja pienten, muutaman koneen yritysten saavuttamattomissa mm. hintansa vuoksi. On myös työmaita, joissa etenemistä voidaan seurata maantieteellisen sijainnin mukaan. Näissä etenemisen seuranta on huomattavasti yksinkertaisempaa. Kun työmaalle asetetaan alku- ja loppusijainti, voidaan työmaan edistymisen mukaan myös ennakoida työmaan valmistumista tai seuraavan vaiheen alkamista. Näin henkilökunnan ja koneiden resurssien ohjauskin voi saada apua pelkän työkoneen sijainnin seurannalla.

Tässä opinnäytetyössä suunnitellaan asiakkaan aloitteesta ensin prototyyppisovelluksen käyttöliittymänäkymä, jossa esitetään merkitty alue tai reitti sekä koneen edistyminen merkityllä alueella tai reitillä. Se on niin sanottu PoC, Proof of Concept eli osoitus projektiin sitoutumisesta. Kaikki seuranta perustuu kartalle merkittyyn tietoon, kuten koneen sijaintiin, sekä tämän tiedon muodostamisen mahdollisuuksien ja resurssien mukaan tehtyihin laskelmiin. Prototyyppisovelluksessa ei vielä ole oikeita toimintoja taustalla, vaan se toimii suunnannäyttäjänä ja havainnollistajana kehittäjille ja asiakkaalle.

Mitä paremmin prototyyppisovellus on asiakkaan mielestä toteutettu, sitä todennäköisemmin asiakas on kiinnostunut käynnistämään varsinaisen kehitysprojektin. Mikäli prototyyppisovellus ei vakuuta asiakasta, kehitysprojektia ei aloiteta. Tehtyä opinnäytetyötä ja siinä luotua prototyyppisovellusta selvitystöineen voidaan kuitenkin käyttää mallina kyseisenkaltaisen projektin mahdollisessa tulevaisuudessa tapahtuvassa toteutuksessa.

2 VAATIMUSMÄÄRITTELY

Tässä opinnäytetyössä suunnitellaan pohja opinnäytetyön kohteena olevalle sovellukselle, hahmotetaan käytettäviä tekniikoita, tehdään alustavia suunnitelmia taustatoiminnoista ja ratkaisusta sekä toteutetaan prototyyppi sovelluksesta. Prototyypillä, joka sisältää aluksi vain perustoiminnot suurpiirteisesti, esitetään asiakkaalle sovelluksen idea ja pyritään hahmottamaan sen toimintaa ja kokonaiskuvaa. Samalla tuodaan esille projektin mahdollisuuksia ja mahdollisesti myös ominaisuuksia, joita ei voida toteuttaa. Prototyyppi toimii myös ideoinnin pohjana alun suunnittelussa asiakkaan kanssa.

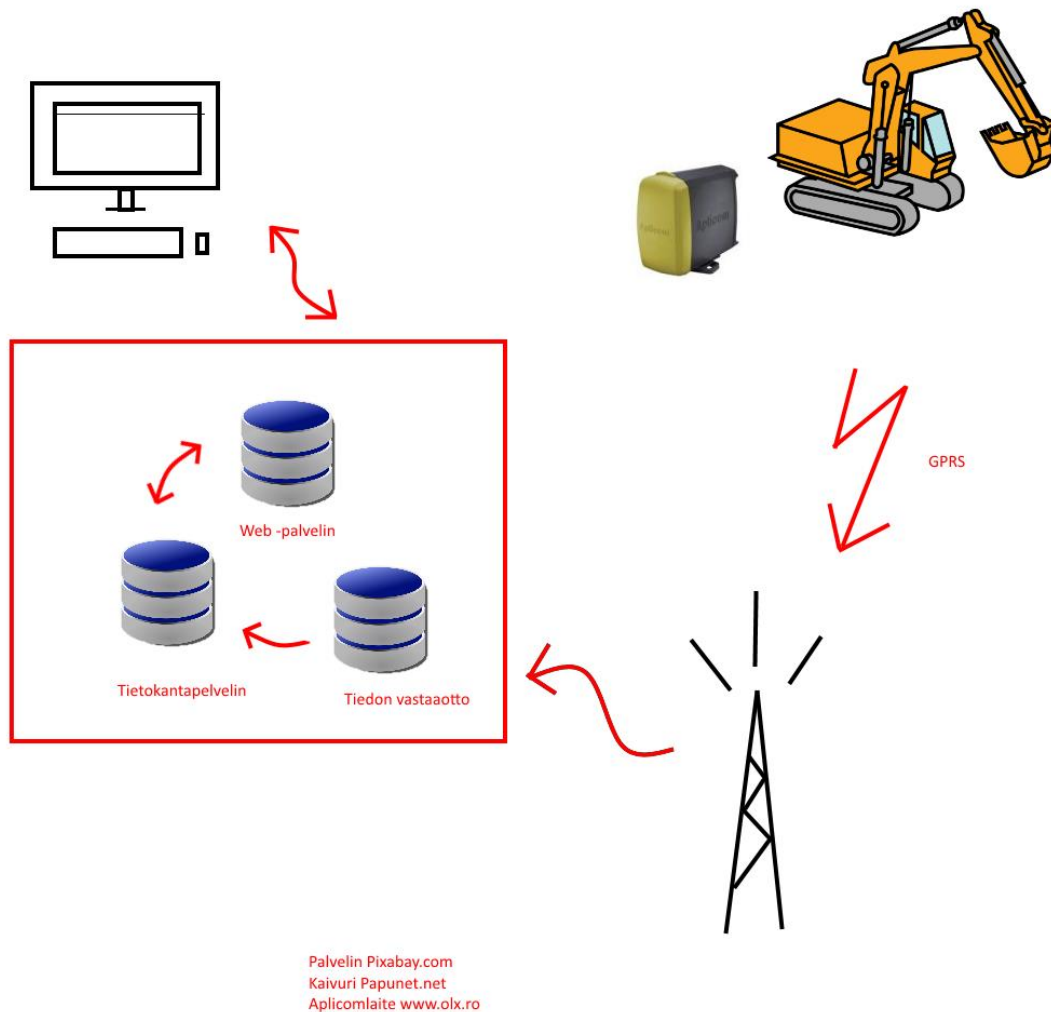
Ennen sovelluksen teon aloittamista hahmotellaan tehtäviä asioita. Näitä ovat mm. kerättävät tiedot, niiden käsittely ja tallennus tietokantaan sekä kannan rakenne. Kuvailaan prototyyppisovelluksen ulkoasu ja siinä esitettävät asiat sekä päätetään käytettävät tekniikat ja alustat. Prototyyppisovelluksessa ja sen esittelyssä kannattaa rajata tehtäviä toimintoja ja esiteltäviä mahdollisuuksia. Tällä pyritään välttämään projektin liiallista monimutkaisuutta ja siten mahdollistaa projektin toteutuminen suunnitellussa aikataulussa.

2.1 Työkoneesta kerättävät tiedot ja tietojen siirto

Suunniteltavassa sovelluksessa seurattavasta koneesta kerättävä tieto on paikkatietoa, joka on nykypäivänä perustietoa ja helposti saatavilla. Saatavan paikkatiedon ei myöskään tarvitse olla yleisesti saatavilla olevaa tietoa tarkempaa. Lähes laite kuin laite, jolla paikkatiedon saaminen ja lähetys eteenpäin on mahdollista, soveltuu tiedon hankkimiseen. Oleellista on tiedon välittyminen palvelimelle siten, että se voidaan liittää johonkin järjestelmässä olevaan koneeseen. Paikkatiedon lähettävästä laitteesta vastaanottavaan palvelimeen tapahtuva tietoliikenne voisi myös olla avointa, salaamatonta tietoa. Osa asiakkaista voi kuitenkin vaatia tiedon salaamista. Siksi lähetettävälle paikkatiedolle on hyvä suunnitella salaustoiminto jo alusta alkaen, mikäli tämä on mahdollista. Sovelluksen web-sivut sisältävät tarkempaa tietoa yrityksistä, joka on siten salasanoin suojattavaa tietoa.

Kuvassa 1 on esitetty tiedonsiirron mallinnus. Ensimmäisenä ketjussa on seurattava työkone. Työkoneeseen on asennettu prototyyppisovelluksessa käytettävä Aplicom-laite, joka lähettää koneen

paikkatiedon mobiiliverkon kautta palvelimelle. Paikkatiedon saamiseksi eteenpäin täytyy olla tiedossa, minne saatu paikkatieto lähetetään. Tiedon vastaanottajana on palvelimen prosessi, joka purkaa vastaanotetun datan sekä tallentaa tai lähettää puretut datat tallennettavaksi tietokantaan. Jotkin toiminnot voivat olla myös sellaisia, että tiedon purkaja välittää tiedon suoraan tietoa käsittelevälle taholle. Tämäkin toiminto on kuitenkin sellainen, että se toteutetaan vasta sitten, kun sille ilmenee tarvetta. Kun tällaisia mahdollisia toimintoja tulee esille suunnittelun ja prototyypitöiden aikana, laitetaan niitä ylös jatkokehitysideoiksi.



KUVA 1. Paikkatiedotodatan siirtyminen työkonessa olevasta Aplicom-laitteesta GPRS-verkon kautta palvelimelle ja sieltä käyttäjän päätelaitteeseen (OLX Aplicom -laitekuva, Papunet työkonokuva, Pixabay Palvelinkuvakkeet).

Alkuvaiheessa seurattavasta koneesta on siis tarkoitus kerätä vain paikkatietoa. Paikkatiedoista on mahdollista päätellä mm. seuraavia tietoja:

- Koneen nykyinen sijainti

- Työkohde, siirto, huoltopiste
 - Osa mainituista toiminnoista edellyttäisi paikannuksen jatkuvaa päällä oloa sekä esim. huoltopisteen määrittämistä.
 - Mahdolliset muut kohteet kuin koneen sijainti työmaalla ovat jatkokehityksen aiheita.
- Koneen paikallaanoloaika nykyisessä sijainnissa (seisonta-aika)
 - Päättellään paikkatiedosta sekä sen viimeisestä saannista.
- Koneen kulkema reitti paikkatiedon tallennuksen aikana
- Koneen etenemistahti
 - Etenemistahdista voidaan ennakoida tiettyyn sijaintiin etenemiseen menevä aika nykyisellä etenemistahdilla.

Sovelluksen paikkatiedon datalähteenä toimii Aplicom A1 BASIX -diagnosointilaitte. Tässä opin-
näytetyössä käytetään paikkatiedon saamiseen laitteeseen integroitua GPS-järjestelmää. Tiedot
välitetään vastaanottavalle palvelimelle laitteessa olevalla matkapuhelinverkon kautta toimivalla
datayhteydellä. Kaikki kerätyt tiedot tallennetaan kantaan, jotta niitä voidaan tarvittaessa käyttää
myöhemmin esille tulevista käyttötapauksista.

Laitteella on mahdollista kerätä koneesta muitakin tietoja. Näitä tietoja voisivat olla kaikki koneen
diagnostiikkaväylästä helposti saatavilla olevat diagnostiikkatiedot. Näitä tietoja voidaan käyttää
esimerkiksi huoltojen suunnitteluun. Myös asiakkaan esittämät toimintoehdotukset käydään läpi ja
toteutetaan, mikäli se yhdessä todetaan kannattavaksi ja halutuksi toiminnoksi. Lisätietojen välittä-
miseen palataan myöhemmin, mikäli siihen tulee tarvetta.

2.2 Web-sovelluksen vaatimukset

Suunniteltavan sovelluksen päätarkoituksena on toimia asiakkaalle työkaluna työkoneiden sijainnin
seurannassa sekä työkoneiden käytön suunnittelussa. Tämä opinnäytetyö sai alkunsa siitä, että

asiakkaalla on ollut ongelmia koneiden sijainnin selvittämisessä ja työmaiden etenemisen hahmottamisessa. Näiden kahden puuttuvan tiedon seurauksena on se, että koneen ja sen kuljettajan seuraavalle työmaalle vapautumisen ennakoiminen on ollut hyvin vaikeaa.

2.2.1 Prototyypisovellus

Prototyypisovelluksella on tarkoitus esitellä alustavasti sovelluksen toimintaa ja näin hahmottaa ja avata sovelluksen tarjoamia mahdollisuuksia. Prototyypisovelluksessa on tarkoitus voida kirjautua sisään annetulla tunnuksilla, jolloin jokaisella on näkyvillä vain itselle tarkoitettu näkymä tietoihin.

Prototyypisovelluksessa voidaan myös karttaan määrittää alue, jossa suunniteltu työ tehdään. Myös koneen sijainti näytetään kartalla. Tiedot tallentuvat kantaan ja näin tietoja voidaan tarkastella myös jälkeen päin. Automaattisia saatuun dataan perustuvia toimintoja ei vielä prototyypisovellukseen ole tarkoitus toteuttaa. Samoin muutkin toiminnot ovat vain pintapuolisia ja tulevaa toimintaa hahmottavia. Esimerkiksi kirjautumisen ei tarvitse toimia oikeasti, ja sivujen tietojen ei tarvitse tulla tietokannasta, vaan ne voivat olla staattisia tietoja, joilla hahmotetaan tulevan sovelluksen toimintaa.

Sovellukset sivuja voisivat olla esim.

- kirjautumissivu
- karttasivu
- kartan editoimissivu
- Käyttäjän tiedot -sivu
- Koneen tiedot -sivu
- Työalueen tiedot -sivu.

Sovelluksessa esitettäviä toimintoja voisivat olla esim.

- sisäänkirjautuminen
- lisättyjen tietojen tallennus
- koneen viimeisen sijainnin näyttö
- koneen kulkema reitti
- työalueen määrittämisen esitysmahdollisuus.

Sovelluksessa esitettäviä asioita ovat koneen sijainti kartalla ja määritellyn työmaan tiedot. Nämä esitetään karttapohjalla. Kullekin esitettävälle asialle on selvät kuvakkeet, joilla pyritään mahdollisimman helposti ymmärrettävässä muodossa esittämään itse asia ja sen tila.

Tarvittavia kuvakkeita prototyypivaiheessa voisivat olla mm.

- Koneen kuvake
 - Erilaisia kuvakkeita koneen eri tiloja esittämään. Esimerkiksi määritellyllä työmaalla, vapaana, huollossa, kuljetuksessa, jne. Tilojen eroina toimivat kuvakkeiden lisäsymbolit ja värit.
 - Erilliset kuvakkeet toimisivat parhaiten, jos niiden vaihtuminen tapahtuisi automaattisesti. Jos tilan vaihto jää kuljettajan tai sovelluksen käyttäjän vastuulle, koneen tila ei ole aina ajan tasalla. Siten siihen ei voida luottaa, ja toiminto jää helposti käyttämättä.
 - Kuvakkeet eri konetyyppien erottamiseen suoraan kuvakkeesta.
- Työmaan kuvakkeet
 - alkusijainti
 - loppusijainti
 - Työmaa-alue, jos toiminto on käytössä.
 - Jollakin värillä muusta alueesta selvästi erottuva alue.
 - Eri kuvakkeita erottamaan työmaatyyppejä toisistaan.
- Muut mahdolliset kohteet kartalla
 - Esim. varikko tai muu huoltopiste.
 - Näiden näyttäminen/piilotus valinnan taakse.

2.2.2 Tietokanta

Tietokantaan tallennetaan kaikki sovelluksessa käytettävä tieto. Tehdäänkö tallennettavalle tiedolle käsittelyä ennen tallennusta vai ei, hahmottunee projektin edetessä. Prototyypivaiheessa käsitteilyä ei vielä ole. Tallennuksen yhteydessä tehtävällä käsittelyllä olisi mahdollista tehdä jo jonkinlaista luokittelua tulevaa tiedonkäsittelyä tai -hakua varten. Tietokannan rakenne määrittyy osaksi organisaatorakenteen ja mahdollisesti esikäsitellyn tiedon mukaan. Organisaatorakenne voi olla sellainen, että jokaiselle organisaatiolle tehdään omia tauluja. Rakenne voi olla myös sellainen, että kaikki samat tiedot tallennetaan yhteen tauluun, jossa ne erotellaan organisaatiotunnuksella. Jos

tietomäärät kasvavat satoihin miljooniin tai isommiksi, voi omien organisaatiokohtaisten taulujen luomisella saada tietokannasta tehtäviä hakuja kevyemmiksi ja nopeammiksi. Organisaatiokohtaiset taulut jäävät jatkokehityksen aiheiksi. Prototyypisovellus toteutetaan tietokannan mallilla, jossa ei ole organisaatiokohtaisia tauluja. Prototyypisovelluksessa tietokantana toimii MariaDB-tietokanta. Valmiissa versiossa käytetään todennäköisesti Microsoftin SQL-tietokantaa ja SQL-palvelintä.

2.2.3 Palvelinalusta

Sovellusta ja tietokantaa tullaan lopullisessa tuotteessa ajamaan palvelimella, jossa on ajossa Microsoftin palvelinperheen tuotteita. Yrityksellä on jo valmiina käytössä palvelimet, joille virtuaaliympäristöt tullaan sitten tilanteen niin vaatiessa asentamaan. Palvelinympäristöjen asennuksen suorittavat ylläpidon henkilöt, kunhan niiden vaatimukset ovat tiedossa ja esimiehiltä tulee pyyntö ja lupa suorittaa asennukset.

2.3 Vaatimukset käytettäville karttapohjan työkaluille

Jotta työmaa voidaan määrittää kartalle, pitää karttanäkymässä olla työkalut työalueen määrittämiseksi. Googlella on valmis työkalupaketti, Drawing Tools, tähän käyttöön. Työkalulla on helppo esitellä tarvittavan kaltaista toiminnallisuutta. Työkaluilla voidaan mm. piirtää alue, tehdä merkintöjä ja merkitä jokin piste kartalle. Nämä työkalut riittävät hyvin sovelluksen tarkoituksiin ainakin tässä suunnittelu- ja prototyypivaiheessa.

Toisena vaihtoehtona työmaa-alueen määrittämiseksi voisi käyttää alku- ja loppupisteiden koordinaattien syöttämistä järjestelmään. Tämän jälkeen loput hoidettaisiin automatisoidusti. Tällöin esim. mutkat työmaa-alueesta tai sen reitistä jäisivät pois. Ensimmäinen vaihtoehto on se, mikä lopullisessa sovelluksessa on tavoitteena, mutta prototyypissä voidaan aikataulullisista syistä joutua luopumaan tästä toiminnallisuudesta. Voidaan jopa kokonaan päätyä staattiseen idean esittämiseen, jossa kohde on ennalta määrätty eikä näin ole muokattavissa.

Kohteen määrittämisestä tallennettavia tietoja ovat määritettävän alueen koordinaatit. Myöhemmin tietoihin lisätään myös tyyppi, mutta se on prototyypissä vasta maininta-asteella. Jos alue voidaan esittää määritettävien pisteiden kautta yhdistämällä ne alueeksi, tarvitaan toiminto, jolla kerrotaan,

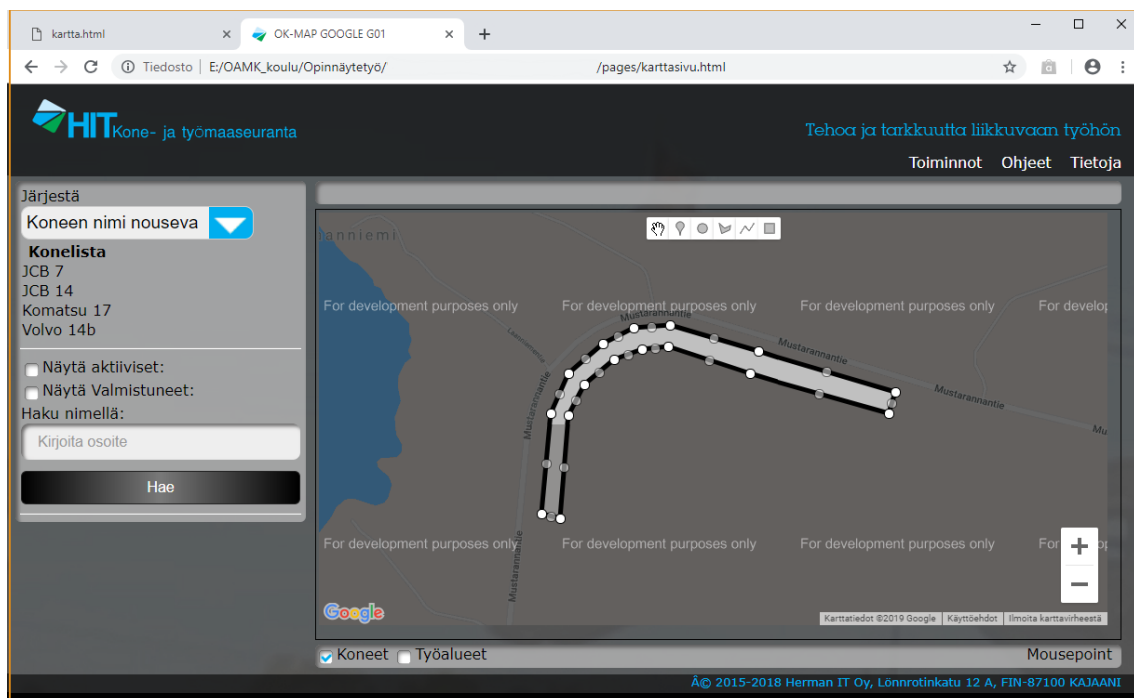
että pisteiden sisälle jäävä alue on määritetty alue. Jos määriteltävä alue on monimuotoinen, esim. pitkä ja mutkikas, voi alueen määrittely muodostua ongelmalliseksi. Määriteltyjen pisteiden sisään jäävä alue voi olla sellainen, että sen määrittäminen yhdessä osassa ei ole mahdollista. Tällöin alue joudutaan määrittämään useassa osassa ja tarvitaan toiminto, jolla kaksi erikseen määriteltyä aluetta voidaan yhdistää yhdeksi alueeksi. Yksi määrittämistä vaikeuttava tekijä voi olla alueiden päällekkäisyys. Toisaalta kaksi erillistä työmaata voivat mennä päällekkäin, mutta alueen määrittelyvaiheessa tehdään vain yhtä aluetta, joten tämä asia ei aiheuta toimenpiteitä. Tämä on asia, joka vaatii enemmän pohdintaa ja selvitystyötä ja jää siten varsinaisen toteutuksen työlisterille. Samoin jo tehdyn kohteen muokkaaminen jäte tään asiaksi, josta on vain mainintoja toimintona.

Kun kohde on määritelty karttaan, pitää se voida nimetä ja tallentaa. Tämä voisi olla yksinkertainen ponnahdusikkuna, jossa olisi kentät tarvittavien tietojen tallentamiseen. Määritellyn työmaan ja määritellyn kohteen yhdistämiseksi tarvitaan myös toiminto. Sen pitäisi olla mahdollisimman yksinkertainen ja helppokäyttöinen. Sillä, kumpi näistä määritellään ensin, ei periaatteessa pitäisi olla merkitystä. Jos ensin on määritelty työmaa, voitaisiin kohdealue sen nimeämisen yhteydessä liittää työmaahan. Vastaavasti työmaan määrittelyssä voitaisiin siihen liittää jo määritelty kohdealue.

3 PROTOTYYPPISOVELLUKSEN TOTEUTUSSUUNNITELMA

Prototyyppisovelluksella oli tarkoitus esitellä alustavasti sovelluksen toimintaa ja näin hahmottaa ja avata sovelluksen tarjoamia mahdollisuuksia. Sovelluksessa on tarkoitus voida näennäisesti kirjautua sisään annetulla tunnuksilla ja määrittää alue, jossa suunniteltu työ tehdään. Myös koneen sijainti näytetään kartalla. Tiedot tallentuvat kantaan ja näin tietoja voidaan tarkastella myös jälkeenpäin. Automaattisia saatuun dataan perustuvia toimintoja ei vielä prototyyppisovellukseen ole tarkoitus toteuttaa.

Sovelluksessa esitettäviä asioita ovat koneen sijainti kartalla ja määritellyn työmaan alue, josta hahmotelma alla olevassa kuvassa 2. Kullekin esitettävälle asialle on selvät kuvakkeet, joilla pyritään esittämään mahdollisimman helposti ymmärrettävässä muodossa kyseinen asia ja sen tila.



KUVA 2. Hahmotelma määritellystä työalueesta

Prototyypin tekeminen aloitettiin suunnittelemalla prototyypin eri osiot. Kuvassa 1 on esitetty tiedonsiirron mallinnus. Ketjussa ensimmäisenä on työkone, jossa on Aplicom-laite, joka lähettää määritellyin aikavälein mittaamansa koneen paikkatiedon mobiiliverkon kautta palvelimelle. Paikkatiedon saamiseksi eteenpäin täytyy olla tiedossa, mihin internetosoitteeseen saatu paikkatieto

lähetetään. Lisäksi tarvitaan käytettävä portti, jolla data saadaan ohjattua halutulle prosessille. Tiedon vastaanottajana on palvelimen prosessi, joka purkaa sen ja tallentaa tiedot tai lähettää ne tallennettavaksi tietokantaan. Nyt tiedot ovat haettavissa web-sovellukseen.

3.1 Aplicom A1 BASIX, datan mittaus ja tiedonsiirto

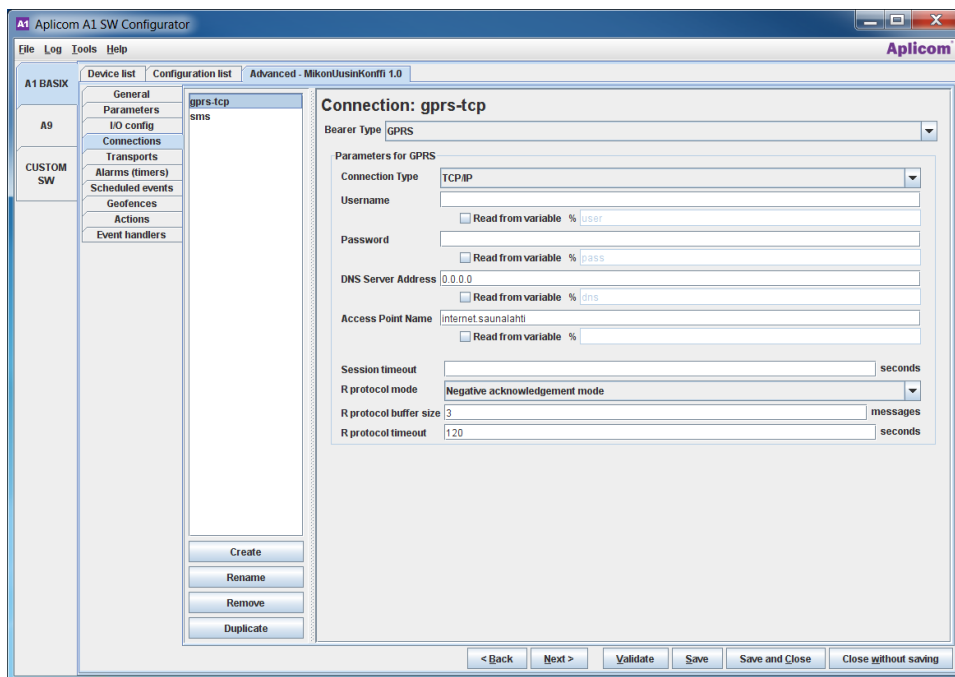
Paikannusdatan siirtoon käytettävä Aplicom A1 BASIX -diagnosointilaitte on tuotteena jo vanhentunut. Niitä on kuitenkin käyttämättömänä yrityksen varastossa, ja niitä voidaan käyttää vapaasti. Laitteen ominaisuudet riittävät paikkatiedon keräämiseen prototyyppisovelluksessa. Työn etenemisen ongelmaksi näytti muodostuvan laitteen konfigurointi. Konfigurointiin tarvittavaa ohjelmistoa oli vaikea löytää. Ottamalla yhteyttä valmistajan tuotetukeen sieltä kuitenkin sai DropBox-linkin, josta ohjelmisto ja siihen liittyvät dokumentit oli mahdollista ladata (1).

Ongelmia oli myös konfiguraatioprofiilin siirtämisessä laitteeseen, kunnes dokumentteja tarkemmin lukiessa selvisi, että siihen pitää olla tietynlainen kaapeli. Varastosta löytyi useampia ulkoisesti vastaavia Aplicom-laitteiden kaapeleita. Tähän piti kuitenkin olla juuri tietty, laitteen ohjelmointiin tarkoitettu kaapeli. Kaapeleissa olivat tyypimerkinnot vielä paikoillaan ja oikea kaapeli löytyi niiden avulla. Toisessa päässä kaapelia on RJ45-tyyppinen liitin, joka liitetään konfiguroitavaan laitteeseen. Kaapelin toinen pää on perinteistä RS232-tyyppiä oleva sarjaporttiin liitettävä pää. Laitteen konfigurointiin tarvittiin siis tietokone, jossa on kyseinen liitäntä. Toinen vaihtoehto oli hankkia USB –adapteri kytkentää varten. Testilaitteen konfigurointiin käytettiin kannettavaa tietokonetta, josta kyseinen sarjaporttiliitäntä sattui löytymään.

3.2 Aplicom A1 BASIXin konfigurointi

Seuraavaksi opeteltiin konfigurointiohjelman toimintaa, että päästiin konfiguroimaan tarvittavat toiminnot ja osoitteet käytettävään laitteeseen. Laitteen konfigurointiin käytettiin Aplicom A1 SW Configurator -ohjelmaa. Tarvittavia tietoja ovat mm. kerättävät paikkatiedot, laitteen tunnustieto, jona toimii laitteen IMEI-koodi (International Mobile Equipment Identity), sekä kohdepalvelimen yhteystiedot. Koska kehitystyötä tehtiin työajan ulkopuolella, datan vastaanottava palvelu oli helpointa pystyttää aluksi omalle tietokoneelle. Näin kehityksenaikaiset ensimmäiset toimintojen testaukset ja tarvittavat korjaukset oli helpompi suorittaa.

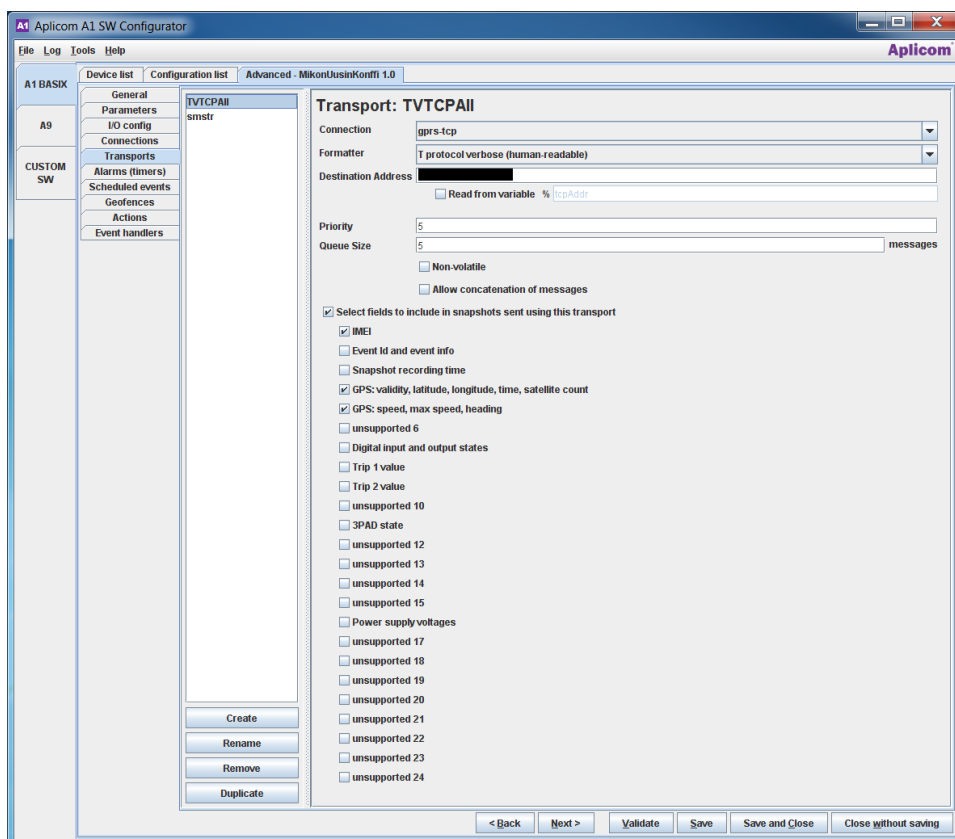
Tietojen lähetysmuodoksi kuvan 3 valikoissa voi Aplicom-laitteesta valita TCP/IP, UDP/IP tai Secure TCP/IP (SSL) -paketin. Datan muodoksi voi valita niin sanotun ihmisen ymmärtämän muodon tai lyhyemmän koneelle kelpaavan muodon. Lisäksi voi vielä valita dataa säästävän paketin, jossa käytetään mahdollisimman vähän dataa tietojen välittämiseen. Prototyypisovelluksessa on valittuna ihmisen ymmärtämä muoto, jossa asiat esitetään purkupäässä perusmuodossaan ja siten koodia on helpompi tehdä ja seurata. Datan määrällä ei ole tässä tapauksessa mitään merkitystä, sillä välitettävien tietojen määrä on hyvin pieni.



KUVA 3. Aplicom A1 BASIX -laitteen tietoliikenneyhteysasetukset

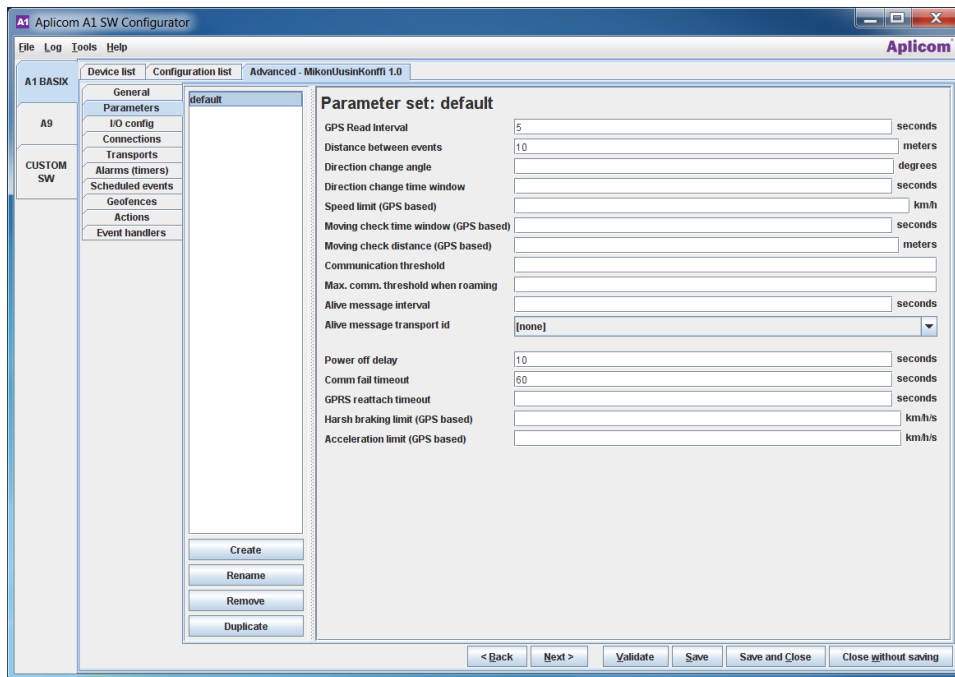
Vastaanotettavia tietoja ovat lähettävän laitteen IMEI sekä LAT- ja LON-koordinaatit. Koordinaatien muodosta päätettiin, että ne ovat ETRS89 TAI WGS84 -tyyppisiä. Laitteen dokumenteista ei löytynyt tarkkaa oikeaa tyyppiä, mutta joissakin koordinaattijärjestelmissä nämä kaksi katsotaan samaksi. Kun nyt tavoitellaan tarkkuutta, joka on ± 10 metriä, niin tämä tarkkuus riittää. Olipa käytetty koordinaattijärjestelmä kumpi tahansa, tästä eteenpäin koordinaateista puhuttaessa tarkoitetaan WGS84-koordinaatteja. Prototyypissä ei paikannuksen tarkkuudella ollut vielä suurta merkitystä, joten kyseinen koordinaatisto on riittävän tarkka tähän tarkoitukseen. Laitteen mittatarkkuus on siirtymisen osalta 15 m, jota pienempiin sijainnin muutoksiin laite ei reagoi. Pienin nopeus, johon laite reagoi, on 5 km/h. Tätä hitaammin liikuttaessa liike ei rekisteröidy nopeudeksi laitteen toiminnoissa.

Konfiguroinnin jälkeen laite on valmiina lähettämään tarvittavia tietoja määritettyyn kohteeseen. Kehitysvaiheessa testejä tehtiin erillisellä tietokoneella kotiolosuhteissa. Datan saamiseksi kotikoneella ajettavaan socket-palveluun täytyi selvittää tietokoneen julkinen IP-osoite. Tämän selvittämiseksi löytyi useita sivustoja, jotka kertovat verkkoon näkyvän IP-osoitteen. Tässä käytettiin osoitetta <https://www.whatismyip.com/ip-address-lookup/>. Kuvan 4 valikoissa IP-osoitteen lisäksi määriteltiin myös porttinumero, johon datapaketti IP-osoitteessa ohjattiin. Tässä esimerkki osoitteen ja porttinumeron esitysmuodosta, jossa osoite on 123.456.789 ja portti on 4321: 123.456.789:4321.



KUVA 4. Aplicom A1 BASIX -laitteen kohdeosoite ja lähetettävien tietojen valikko.

Asetuksia ei saatu aivan halutunlaisiksi. Kuvassa 5 oleviin asetuksiin kokeiltiin erilaisia yhdistelmiä, mutta silti ei saatu toivottua lopputulosta. Paikkatietodatan muutosten lähettäminen määritettyjen ehtojen mukaan ei tapahdu halutulla tavalla. Nyt tieto tulee tietyin aikavälein, vaikka aikaväliä ei ole määritetty tai se on määritetty haluttua lyhyemmäksi. Myös matkan muutoksiin perustuvat toiminnot toimivat toisin, kuin mihin määrittelyillä on pyritty. Data tulee joko liian harvoin suhteessa matkaan tai se tulee aikaperustaisesti. Prototyypivaiheessa paikkatietojen lähetys olisi haluttu lyhyemmin matkavälein tai tietyin aikavälein yhteyksien toiminnan seuraamiseksi.



Kuva 5. Aplicom A1 BASIX -laitteen parametrien asetusvalikko.

Kuvassa 5 olevien parametrien selvennyksiä:

- GPS Read Interval: kuinka usein laite lukee GPS-paikkatiedon, yksikkö sekuntia
- Distance between events: Matka, jonka jälkeen generoidaan uusi paikkatieto, yksikkö metriä
- Speed limit: Nopeusraja, jonka ylittyessä generoidaan tarvittaessa ilmoitus nopeudesta, yksikkö km/h
- Moving check distance (GPS based): Matka, jonka ylittyessä generoidaan viesti muuttuneesta etäisyydestä, yksikkö metriä
- Alive message interval: Aika, jonka jälkeen generoidaan viesti laitteen päällä olemisesta, jos muita viestejä ei ole generoitunut määritetyn ajan sisällä, yksikkö sekuntia.

Kuvassa on nähtävillä myös muita parametrejä, mutta koska haussa olivat vain paikannustiedot, ei muita parametrejä ole määritetty. Käytetyt parametrit lähettävät viestin niiden raja-arvojen ylittyessä tai täytyessä. Osaa parametreista, kuten Moving check distance, ei paikannukseen tarvita. Koska halutun kaltaista toimintaa paikannukselle ei saatu, testattiin paremman lopputuloksen aikaansaamiseksi muitakin parametrejä.

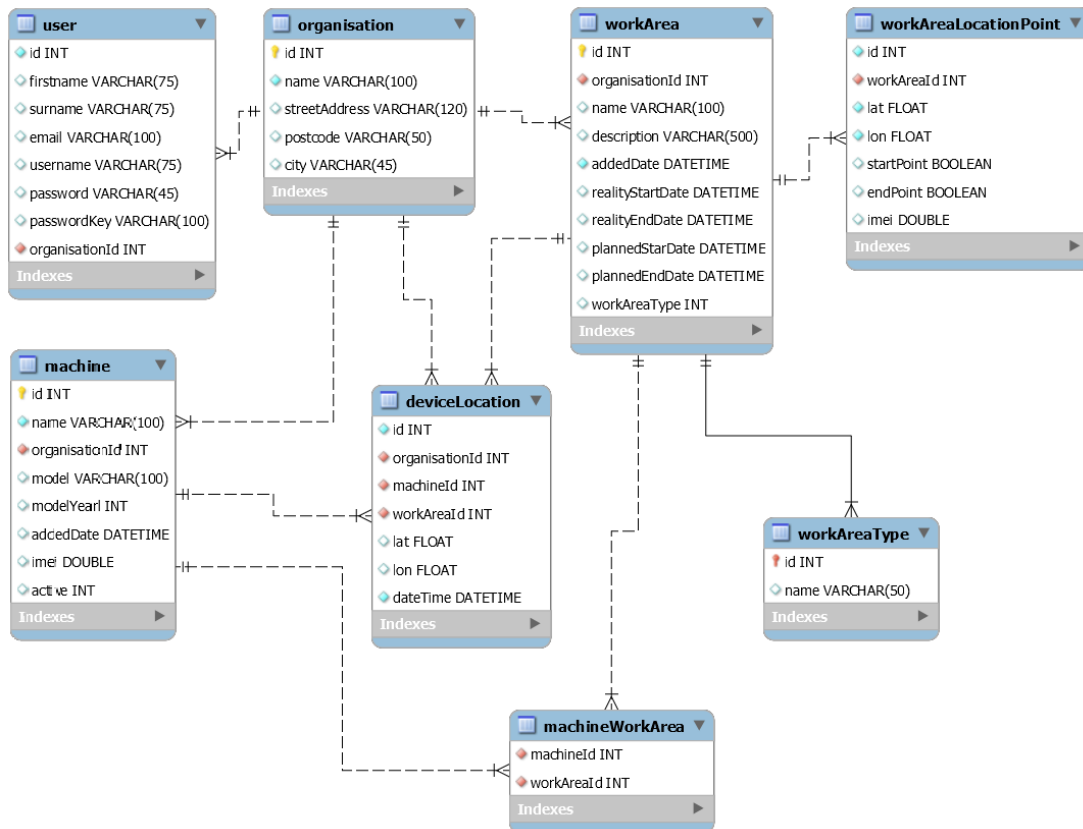
3.3 Tietojen vastaanotto ja vastaanotetun datan purkaminen

Aplicom-laitteen lähettämän TCP-IP -muotoisen datapaketin vastaanottajana toimii palvelimella oleva ohjelma, jossa on socket-palvelu tätä varten. Palvelu kuuntelee määriteltyä porttia ja vastaanottaessaan datapaketin kyseiseen porttiin purkaa sen sisältämän datan. Mikäli halutun tyyppiä avain-arvopareja ei löydy, purettu data hylätään ja siirrytään odottamaan seuraavaa datapakettia. Datapaketista prototyyppivaiheessa eriteltäviä tietoja ovat laitteen IMEI-koodi ja GPS-koordinaatit (liite 1.). Datan purkamisen jälkeen tiedot tallennetaan tietokantaan saadun IMEI-tiedon perusteella. IMEI-tiedolla vastaanotettu data saadaan liitettyä muihin kannan tietoihin ja tallennettua oikeilla tiedoilla oikeaan paikkaan. Socket-palvelussa on käytetty pohjana Microsoftin manuaaleista löytyviä esimerkkejä, joita on muokattu tässä projektissa toimiviksi (3.).

3.4 Tietokanta

Vastaanotetusta datasta puretut tiedot tallennetaan tietokantaan, jonka toimintakuntoon saamiseksi löytyi hyviä ohjeita useammastakin lähteestä (3; 4; 5). Kun tietoja myöhemmin tarvitaan, voidaan ne hakea tietokannasta. Jotta voitaisiin esittää oikea tieto oikeassa asiayhteydessä, täytyy tallennetavalle tiedolle suunnitella rakenne, joka soveltuu tähän tarkoitukseen. Tässä sovelluksessa on tarkoitus käyttää Microsoftin SQL-tietokantaa. Prototyyppisovelluksessa käytetään kuitenkin toteutussyistä MariaDB-tietokantaa, jota ajettiin XAMPP-ohjelmaan kuuluvalla tietokantapalvelimella.

Olennaista tietokannassa on sen rakenteen suunnittelu tarpeen mukaan. Tällöin tietokannalla on mahdollisuus toimia optimaalisella tavalla ja sillä on mahdollisuus suoriutua helposti suurien tietomäärien käsittelystä. Tässä sovelluksessa käytetään perinteistä relaatiotietokantarakennetta, koska se soveltuu parhaiten projektissa syntyvän tiedon tallentamiseen. Tietokannan EER-kaavio on esitetty kuvassa 6.



KUVA 6. Prototyypisovelluksen tietokantahahmotelma

Tietokannan keskeisiä tietoja ovat organisaatio (organisation), kone (machine) ja työalue (workArea). Nämä tiedot ovat linkitettyinä toisiinsa joko suoraan tai välillisesti. Uuden tietokannan luomisen jälkeen riittää kuitenkin, että organisaatio-taulussa on yksi organisaatio ja sille käyttäjä-taulussa linkitetty käyttäjä (user). Kun nämä kaksi tietoa ovat määriteltyinä, voi sisään kirjautunut käyttäjä luoda organisaatiolle koneen ja työalueen. Jotta koneen paikannustiedot voitaisiin tallentaa kantaan, täytyy koneelle määritellä järjestelmään siihen asennetun paikannuslaitteen IMEI-koodi. Lisäksi kyseinen kone täytyy liittää halutulle työalueelle tai valita työalueelle haluttu kone. Nyt koneesta vastaanotettu data voidaan ohjata siinä olevan IMEI-koodin avulla oikealle organisaatiolle ja edelleen oikealle työalueelle laitteen sijainti-tauluun (deviceLocation).

Yksi kone voi kerrallaan olla vain yhdellä työalueella, mutta yhdellä työalueella voi olla kerrallaan useita koneita. Siksi täytyi luoda ns. linkkitaulu kone- ja työaluetiedon yhdistämiseksi. Työmaan tyyppi-taululle (workAreaType) ei vielä ole tarkemmin suunniteltua käyttöä. Se otettiin kuitenkin mukaan, koska ennakoitiin myöhemmin olevan tarvetta erotella työalueita myös niiden tyypin perusteella. Kaivinkoneen työaluetyppejä voisivat olla esimerkiksi kaapelityö, putkityö (viemäriputkisto, kaukolämpö, jne.) ja tietyö.

Mahdollisen historiatiedon tallennusta ei ole otettu prototyypitietokannan suunnittelussa laajemmin huomioon. Yhtenä historiatietona voisi toimia mm. koneen aktiivinen/passiivinen-tilan määrittäysmahdollisuus. Tämän tiedon avulla olisi mahdollista nähdä, mitä koneita on ollut aikaisemmin käytössä. Siten aiemminkin valmistuneella työalueella käytetyt koneet tietoineen voidaan hakea esitetystä tietokantarakenteesta.

4 TYÖN TILANNE OPINNÄYTETYÖN PÄÄTÖSVAIHEESSA

Opinnäytetyön alkuperäisenä tavoitteena oli tehdä prototyyppisovellus, jolla hahmoteltaisiin haluttuja toimintoja ja saataisiin asiakkaalle käsitys suunniteltavan sovelluksen mahdollisuuksista ja hyödyistä. Samalla saataisiin itse tarkempi käsitys toteutuksen käytännön vaatimuksista. Työn toteutus venyi odotettua pidemmäksi tekijän aikaisemman kokemattomuuden, jaksamisen ja muiden töiden vuoksi. Työn aloittamisesta valmiiksi saamiseen meni aikaa yli vuosi. Työn haasteellisuus ja aikaisemman kokemuksen puute tuli vastaan työn useiden eri osa-alueilla joista tekijällä ei ollut aikaisempaa osaamista. Tämän vuoksi varsinainen asiakkaalle esittely jäi tekemättä, koska asia eteni prototyyppivaiheen ohi toisella taholla. Asiakkaan kanssa käydyissä keskusteluissa oli tullut esille myös muita sovellukseen mahdollisesti hahmoteltavia toimintoja. Näin alkuperäinen tavoite esitellä sovellusta asiakkaalle tämän työn tuloksena syntyneellä hahmotelmalla jäi tekemättä. Sen sijaan melkein kaikkia tehtyjä hahmotelmia, suunnitelmia ja valmiiksi saatuja toimintoja voidaan käyttää lähes sellaisenaan mahdollisissa muissa projekteissa.

4.1 Taustatoimintojen vaiheet

Sovellukseen saatiin toteutettua tietojen keräyksen ja kantaan tallennuksen toiminnot. Nämä ovat lähes sellaisenaan käytettävissä edelleen kehittämissä projektisuunnitelmissa. Varsinkin Aplicom-laitteen asetusten teko on palautettu mieleen, ja siihen on nyt valmiit työkalut. Tosin työssä käytetty laitteen vanhentunut versio ei sellaisenaan sovellu kuin esittelykäyttöön. Tietojen vastaanotto toiminee uusilla laitteilla samoilla periaatteilla, joten sitä voi käyttää jatkossa muissakin projekteissa, joissa tarvitaan samankaltaisia toimintoja. Toiminto on toteutettu itsenäisenä prosessina, joten sen siirtäminen toisiin projekteihin käy hyvin helposti.

Myös tietokannan rakenne sijaintitietojen tallennusta varten toimii tämänkaltaisissa tarpeissa pienin muutoksin. Tämä toteutuu varsinkin silloin, kun dataa lähettävä laite saadaan liitettyä muuhun järjestelmään laitteen IMEI-koodilla, kuten se on tässäkin projektissa tehty. Tietojen siirto tietokannasta web -sivulle sen sijaan jäi toteuttamatta jo aikaisemmin esille tulleiden tekijään liittyvien- ja muiden tekijöiden vuoksi.

4.2 Web-sivut

Prototyypisovelluksen web-sivuilla tarkoitus oli saada asiakas kiinnostumaan sovelluksen kehityksen aloittamisesta ja siten saada työntajalle uusi asiakasprojekti. Sivuihin oli jo projektin ensimmäisten kuukausien jälkeen luotu jonkinlaiset karkeat mallit. Tietotaito ja käytettävissä oleva aika kuitenkin pysäyttivät tämän osion etenemisen tälle tasolle, ja sivuja ei koskaan päästy esittelemään asiakkaalle.

Karttaan lähes ainoaksi tavoitelluksi toiminnoksi suunniteltu sen alueen tai reitin merkintä, jota kone tulee työmaalla etenemään, osoittautui tekijälle liian haastavaksi tehtäväksi. Vaikka Googlella oli tarjota lähes valmiita työkaluja, ei haluttujen toimintojen lisäys onnistunut ongelman ratkaisemiseksi käytetyistä kymmenistä työtunneista huolimatta. Liitteissä 2 ja 3 on kuvia karkeista sivuhahmotelmista.

4.3 Oppimiskokemukset

Koneiden- ja työalueiden etenemisen seurantaan käytettävä sovellus oli tekijästä aiheena mielenkiintoinen. Jo aloituspalaverissa tuli selville, että työssä tarvittaisiin useita eri tekniikoita. Osasta käytettäviä tekniikoita oli tekijällä hyvin vähän aikaisempaa kokemusta. Näiden teon arveltiin voivan osoittautua haasteelliseksi. Lopulta kävikin niin, että projekti eteni muilla tahoilla ohi tämän työn tarpeen. Etenkin Web -ohjelmointi oli tekijälle uutta, ja sen tuomat haasteet viivyttivät omalta osaltaan työn toteutusta. Kaikkiaan projektissa aikaan saadut hahmotelmat sivuista jäivät toteutuksen osalta puolitehen. Olisi ollut mielenkiintoista esitellä prototyypisovellus asiakkaalle ja saada suoraa palautetta tekemästään työstä. Työssä oli kuitenkin paljon mielenkiintoisia asioita, joita tehdessä tuli onnistumisen elämyksiä. Toisaalta oli myös asioita, jotka olisi ollut kiva saada toteutettua toimiviksi asti. Monia asioita ja pohdintoja, joista osa ei päätynyt tähän työhön, on jo käytetty muualla projektin kehityksessä.

Tehtyjen suunnitelmien perusteella työläin osa-alue prototyypisovelluksessa tulee varmaan karttapohjalla toimivien toimintojen toteutus. Vaikka Googlelta löytyykin kirjasto, jossa on useita toimintoja lähes käyttövalmiina, haluttujen toimintojen toteutus vaatii omaakin kehitystä runsaasti. Sovelluksen muut sivut, kuten koneen tiedot ja työmaantiedot -sivut, joilla mahdollistetaan karttasivun

toiminnot, eivät itsessään ole monimutkaisia toteuttaa. Nekin vaativat kuitenkin oman aikansa, ennen kuin ovat käyttövalmiina.

LÄHTEET

1. Aplicom tuotetuki 2018. Opinnäytetyötä varten latauslinkin kautta saatu materiaali Aplicom BASIX A1 tuotteesta. Ladattu 14.3.2018.
2. Microsoft 2017. Asynchronous Server Socket Example. Saatavissa: <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/asynchronous-server-socket-example>. Hakupäivä 24.3.2019.
3. Etienne Rached 2009. Connect C# to MySQL. Saatavissa: <https://www.codeproject.com/Articles/43438/Connect-C-to-MySQL>. Hakupäivä 24.3.2019.
4. MySQL 2019. Begin Your Download. Saatavissa: <https://dev.mysql.com/downloads/file/?id=484706>. Hakupäivä 24.3.2019.
5. Stackoverflow 2014. How to connect to MySQL Database? Saatavissa: <https://stackoverflow.com/questions/21618015/how-to-connect-to-mysql-database> (keskustelupalsta). Hakupäivä 24.3.2019.

Koodi, jolla laitetunnus (IMEI) ja LAN ja LON koordinaatit selvitetään vastaanotetun datan joukosta.

```
string[] lineWordsSeparated = tmpLine.Split(new string[] { ":" }, StringSplitOptions.None);
if (lineWordsSeparated[0].ToUpper().Equals("LAT"))
{
    try
    {
        dataValues.LAT = float.Parse(lineWordsSeparated[1]);
    }
    catch (Exception e)
    {
        //Parsing LAT failed. Lets replace decimal mark and try again.
        try
        {
            dataValues.LAT = float.Parse(lineWordsSeparated[1].Replace('.', ','));
        }
        catch (Exception e)
        {
            // Failed to parse LAT
        }
    }
}
else if (lineWordsSeparated[0].ToUpper().Equals("LON"))
{
    try
    {
        dataValues.LON = float.Parse(lineWordsSeparated[1]);
    }
    catch (Exception e)
    {
        //Parsing LON failed. Lets replace decimal mark and try again.
        try
        {
            dataValues.LON = float.Parse(lineWordsSeparated[1].Replace('.', ','));
        }
        catch (Exception e)
        {
            // Failed to parse LON
        }
    }
}
else if (lineWordsSeparated[0].ToUpper().Contains("IMEI"))
{
    try
    {
        dataValues.IMEI = long.Parse(lineWordsSeparated[1]);
    }
    catch (Exception e)
    {
        // Parsing IMEI failed.
    }
}
```

OK-MAP GOOGLE G01 x +

Tiedosto | E:/OAMK_koulu/Opinnäytetyö/ /pages/konesivu.html ☆ | |

HIT Kone- ja työmaaseuranta Tehoa ja tarkkuutta liikkuvaan työhön

Toiminnot Tietoja

Järjestä
Koneen nimi nouseva ▼

Lisää uusi kone

Hae paikkakunnittain:
Kajaani ▼

Hae

Koneen malli ja ominaisuus tiedot

Koneen merkki:
Kirjoita merkki

Koneen malli:
Kirjoita malli

Koneen vuosimalli:
Kirjoita vuosimalli

Kotikunta :
Kirjoita kunta

Paikannuslaitteen IMEI:
Kirjoita IMEI

© 2015-2018 Herman IT Oy, Lönnrotinkatu 12 A, FIN-07100 KAJAANI

OK-MAP GOOGLE G01

Tiedosto | E:/OAMK_koulu/Opinnäytetyö/ /pages/kayttaja.html

HIT Kone- ja työmaaseuranta Tehoa ja tarkkuutta liikkuvaan työhön

Toiminnot Tietoja

Järjestä
Nimi nouseva

Lisää käyttäjä

Hae paikkakunnittain:
Kajaani

Hae

Käyttäjän tiedot

Etunimi:
Kirjoita etunimi

Sukunimi:
Kirjoita sukunimi

Puhelinnumero:
Kirjoita puhelinnumero

Kotikunta :
Kirjoita kotikunta

© 2015-2018 Herman IT Oy, Lönnrotinkatu 12 A, FIN-87100 KAJAANI