



**LAUREA**  
AMMATTIKORKEAKOULU  
*Yhdessä enemmän*

# Sää reitillä -palvelun kehittäminen

Kuoremäki, Mikko

2015 Laurea Leppävaara



Laurea-ammattikorkeakoulu  
Leppävaara

## Sää reitillä -palvelun kehittäminen

Mikko Kuoremäki  
Tietojenkäsittelyn koulutusohjelma  
Opinnäytetyö  
Helmikuu, 2015

Laurea-ammattikorkeakoulu  
 Laurea Leppävaara  
 Tietojenkäsittelyn koulutusohjelma

Tiivistelmä

Mikko Kuoremäki

### Sää reitillä -palvelun kehittäminen

Vuosi	2015	Sivumäärä	25
-------	------	-----------	----

Tämän toiminnallisen opinnäytetyön tarkoituksena oli toteuttaa uusi interaktiivinen Sää reitillä -palvelu Ilmatieteen laitoksen Ilmanet palvelukanavaan. Opinnäytetyön tavoitteena oli luoda uudenlainen palvelu, jossa asiakkaat pystyisivät helpommin saamaan tarkempaa säätietoa valitsemalleen reitille.

Opinnäytetyön toimeksiantajana oli Ilmatieteen laitos. Työ toteutettiin jo valmiina olevaan Ilmatieteen laitoksen Ilmanet asiakas palvelukanavaan lisäpalveluna.

Opinnäytetyön tietoperustana perehdyttiin sivuston luomiseen ja sivuston reponsiiviseen toiminnallisuuteen. Opinnäytetyössä tutustuttiin kirjallisuuteen sivuston toteutus menetelmistä, jotka kartoitettiin palvelun eri osien rakentamista varten.

Tutkimusmenetelminä käytettiin dokumenttianalyysiä ja workshopityöskentelyä. Dokumentteista, jotka on tehnyt ulkopuolinen yritys asiakaskyselyinä, saatiin asiakasvaatimukset. Yrityksen sisäisessä workshopissa vaatimukset arvioitiin, preferoitiin, täydennettiin ja muokattiin palvelun mahdollisiksi ominaisuuksiksi.

Palvelu suunniteltiin ja toteutettiin asiakastoiveiden ja workshopin määritysten mukaan. Projektista vastasi pääsääntöisesti opinnäytetyön kirjoittaja yksin, mutta sai tukea tarvittaessa ongelma tilanteissa muilta tiimin jäseniltä.

Opinnäytetyön tavoite saavutettiin ja palvelu saatiin valmiiksi ennalta sovituin ominaisuuksin määräaikaan mennessä. Palvelu on tällä hetkellä testikäytössä muutamalla asiakkaalla, ja palaute siitä on ollut positiivista.

Asiasanat      Interaktiivinen, sää, karttapalvelu

Mikko Kuoremäki

### Weather on the Route -Service Development

Year	2015	Pages	25
------	------	-------	----

The purpose of this thesis was to implement a new interactive Weather on the Route service to the Finnish Meteorological Institute's Ilmanet service channel. The objective was to create a new kind of service to the customers in which customers would be able to obtain more easily more detailed weather information for their chosen routes.

The Finnish Meteorological Institute commissioned this thesis. The work was carried out to an already existing Finnish Meteorological Institute's customer service channel as an additional service.

The theoretical part of this thesis focuses on the site creation and its responsive functionality. This thesis explored the literature of site implementation methods, which were charted for the development of different parts of the service.

The research methods used were document analysis and working in a team workshop. From the documents that have been made by an external company as customer surveys were gathered the customer requirements. In company's internal workshop the requirements were estimated, preferred, supplemented and modified to obtain potential features for the service.

The service was planned and carried out by the wishes of the customers and specifications from the workshop. In general, the project was carried out by the thesis writer alone, but with the support of other team members in problematic situations.

The objective of this thesis was achieved and the service was completed with pre-agreed functionality by the time of deadline. The service is currently in test use by a few customers, and the feedback from them has been positive.

Keywords     Interactive, weather, map service

## Sisällys

1	Johdanto .....	6
2	Työn tavoitteet ja menetelmät .....	6
3	Keskeiset käsitteet .....	6
4	Ilmatieteen laitos.....	7
5	Palvelun määrittely ja suunnittelu .....	8
5.1	Interaktiivinen kartta .....	8
5.2	Responsiivinen taulukko.....	9
6	Palvelun toteutus.....	9
6.1	Kartan rakentamien .....	10
6.1.1	Kartan perusominaisuudet .....	10
6.1.2	Reitti ominaisuudet.....	10
6.1.3	Kartan hallinta liittymän suunnittelu ja toteuttaminen .....	11
6.1.4	Sää tietojen esittäminen kartalla .....	11
6.1.5	Koordinaatio muunnokset karttaan piirrettäessä .....	12
6.1.6	Kartta kokonaisuus.....	13
6.2	Taulukon rakentamien.....	14
6.3	Kokonaisuuden rakentaminen .....	16
6.3.1	Ilmanet .....	16
6.3.2	Kartan integroiminen .....	16
6.3.3	Taulukon integroiminen.....	16
6.3.4	Datahaut.....	16
6.3.5	Kokonaisuuden toiminnan yhdistäminen .....	17
6.4	Responsiivisuus .....	18
6.4.1	Kartan responsiivisuus.....	19
6.4.2	Taulukon responsiivisuus.....	20
6.5	Tilauksen muokattavuus.....	21
7	Sää reitillä -palvelu .....	22
8	Loppuarviointi.....	23
	Lähteet .....	24
	Kuvat .....	25

## 1 Johdanto

Opinnäytetyön aiheena oli interaktiivisen sääpalvelun toteuttaminen merenkulkijoille. Palvelussa käyttäjät voivat itse määrittellä reittinsä, jolle saavat säätietoja. Palvelun nimeämisessä otettiin huomioon palvelun tarkoitus. Sää reitillä oli kuvaavin nimi palvelulle, koska palvelu tarjoaa säätietoja valituille reiteille, mutta ei ota kantaa miten reittejä tulisi suunnitella. Opinnäytetyö toteutettiin Ilmatieteen laitokselle, joka on nykyinen työnantajani.

## 2 Työn tavoitteet ja menetelmät

Opinnäytetyön tavoitteena oli toteuttaa interaktiivinen Sää reitillä -palvelu Ilmatieteen laitoksen Ilmanet asiakas palvelukanavaan. Valmiin palvelun on tarkoitus täydentää potentiaalisten asiakkaiden tuotevalikoimaa ja jopa korvata useamman yksittäisen tuotteen tarvetta tuomalla kattavamman yhden tuotteen kokonaisuuden, joka toisi mahdollisimman paljon lisäarvoa asiakkaalle.

Tietoperustana perehdyttiin internet sivustojen luomiseen ja niiden reponsiiviseen toiminnallisuuteen. Painottaen responsiivisuutta, koska sen avulla palvelun käytettävyyttä saadaan parannettua ja käyttömahdollisuuksia laajennettua.

Tutkimusmenetelminä käytettiin dokumenttianalyysiä ja workshopityöskentelyä. Dokumentteista, jotka on tehnyt ulkopuolinen yritys asiakaskyselyinä, saatiin asiakasvaatimukset. Yrityksen sisäisessä workshopissa vaatimukset arvioitiin, preferoitiin, täydennettiin ja muokattiin palvelun mahdollisiksi ominaisuuksiksi.

Palvelu toteutettiin näistä saatujen määritysten perusteella pääasiallisesti opinnäytetyön kirjoittajan toimesta, ongelma tilanteissa neuvoja saatiin tiimin asiantuntijoilta.

## 3 Keskeiset käsitteet

Ilmanet on Ilmatieteen laitoksen yritysasiakkaille räätälöity palvelukanava. Ilmanetin kautta saat uusimmat säätiedot selkeinä kuvina tai tekstimuotoisina tietopaketteina. (Ilmatieteen laitos 2015)

API (Application Programming Interface) tarkoittaa ohjelmistorajapintaa, jonka avulla verkossa toimivaan sovellukseen voidaan tehdä ohjelmallisia kutsuja esimerkiksi toisesta ohjelmistosta. (Cenno 2014)

HTML (Hypertext Markup Language) on kuvauskieli Web-dokumenttien kuvaamista varten. (w3school 2015)

JavaScript on Web:in ohjelmointikieli. Kaikki modernit HTML-sivut käyttävät JavaScript:iä. (w3school 2015)

OpenLayers on puhdas JavaScript -kirjasto karttatiedon esittämiseen useimmissa moderneissa web-selaimissa ilman palvelinpuolen riippuvuuksia. (OpenLayers 2008)

CSS (Cascading Style Sheet, porrastetut tyyliarkit) on erityisesti WWW-dokumenteille kehitetty tyyliohjeiden laji. CSS:ssä dokumenteille voi määritellä useita tyyliohjeita, jotka yhdistetään tietyllä tavalla yhdeksi säännöksi. (web-opas 2015)

Responsiivisuudella tarkoitetaan yksinkertaisesti mukautuvuutta. Responsiivisesti toteutettu sivusto tunnistaa käytettävän laitteen ja mukauttaa sisällön, ulkoasun ja erilaiset toiminnallisuudet automaattisesti käytettävän laitteen mukaan, joten erilliset mobiilisivustot ja saman sisällön päivittämiset useaan paikkaan on onneksi historiaa. (Responsiivisuus info 2015)

GeoServer on avoimen lähdekodin palvelin jakamaan paikkatietoja. Suunniteltu yhteentoimivuus, se julkaisee dataa kaikilta merkittäviltä paikkatiedon tuottajilta käyttäen avoimia standardeja. (GeoServer 2014)

WMS (Web Map Service) on standardi protokolla palvelemaan georeferoituja karttakuvia, joita kartta palvelin tuottaa. Lyhyesti WMS on tapa jolla asiakas voi pyytää kartanosia serveriltä. (OpenGeo 2015)

JSON (JavaScript Object Notation) on syntaksi tiedon tallentamiseen ja jakamiseen. (w3school 2015)

PHP on suosittu yleiskäyttöinen skriptikieli, joka sopii erityisesti web-kehitykseen. (PHP 2015)

#### 4 Ilmatieteen laitos

Ilmatieteen laitos on palvelu- ja tutkimuslaitos, joka tuottaa yleisen turvallisuuden ja elinkeinoelämän toimintaedellytysten kannalta tärkeitä sää-, meri- ja ilmastopalveluja.

Ilmatieteen laitoksen palvelut pohjautuvat vahvaan ilmakehä- ja meriaiheiseen tutkimukseen ja korkealaatuisiin havaintoihin. Ympäri vuorokautinen toiminta tukee viranomaisten, yritysten ja kansalaisten toimintaa myös silloin, kun sää, meri ja luonto tuovat eteen erityishaasteita.

Ilmatieteen laitos on perustettua vuonna 1838. Se on hallinnollisesti liikenne- ja viestisätintäministeriön alainen ja toiminut nykyisellä nimellä toiminut vuodesta 1969. (Ilmatieteen laitos 2015)

## 5 Palvelun määrittely ja suunnittelu

Ilmatieteen laitoksen teettämästä asiakaskyselystä ilmeni tarve helppokäyttöiselle palvelulle, jota on selkeä käyttää. Säättietojen tarkempi saaminen reitille oli kyselyissä toivotuin palvelukokonaisuus. Esimerkkeinä oli mainittu vastaavanlaisia, mutta eri tarkoituksiin toteutettuja interaktiivisia palveluita. Näiden perusteella tiimin workshopissa koottiin ominaisuudet, jotka tulevaan palveluun toteutettaisiin.

### 5.1 Interaktiivinen kartta

Suunnittelun ensimmäisenä vaiheena oli löytää sopiva kartta palveluun. Kartta on keskeinen komponentti palvelussa, joten sen pitäisi toimia kaikissa tapauksissa moitteettomasti. On olemassa monia potentiaalisia kartta -kirjastoja minkä päälle palvelun kartan voi rakentaa. Karttakirjaston kriteerit kirjattiin workshopissa seuraaviksi:

- Responsiivinen toimivuus
- Viivojen piirto ja muokkaus ominaisuus
- Painettavien markkereiden laittaminen
- Uusimpien mobiililaitteiden tuki
- Internet selainten laaja tuki
- WMS-layer tuki

Testaukseen valikoitui seuraavat JavaScript -kirjastot: OpenLayers, Leaflet ja Google Maps. Parhaiten projektiin sopivimmat ominaisuudet löytyi OpenLayers kartta -kirjastosta. Sen uusin versio 3 täytti kaikki vaatimukset halutulle kartta -kirjastolle.

OpenLayers:ltä löytyi hyvät esimerkit kaikista tarvittavista ominaisuuksista heidän kotisivuiltaan, jossa pystyttiin testaamaan kaikki tarpeelliset ominaisuudet, tarvitsematta käyttäen aikaa niiden toteuttamiseen. Osana valintaa laajojen esimerkkien lisäksi oli myös kattava dokumentaatio kartta -kirjaston API:sta. Myös aikaisemmat kokemukset OpenLayers kartta -kirjaston edeltävästä versiosta helpottivat valintaa ja tietämys miten laaja käyttäjien luoma verkosto sillä on, jonka kautta voi löytää tai saada apua.



Leaflet ja Google Maps kirjastot olivat suhteellisen hyviä, mutta piirto-ominaisuudet eivät olleet yhtä kattavat, kuin mitä oli OpenLayers:ssä.

## 5.2 Responsiivinen taulukko

Responsiivinen taulukko oli toinen vaatimuksista. Pääasiallisena vaatimuksena oli, että se toimii responsiivisessa ulkoasussa.

Kartan alle pääasiallisen datan esittämiseen tarkoitettulla taulukolla ei niin laajoja vaatimuksia ollut, kuin mitä kartalla. Pääasiallisena vaatimuksena oli vain, että se toimii responsiivisessa ulkoasussa.

Alustavasti tähän valittiin DataTables nimisen JavaScript kirjaston, jonka avulla tavallisen HTML taulukon pystyy muuntamaan responsiiviseksi. Se olisi toiminut seuraavalla periaatteella: JavaScript kirjasto seuraa taulukolle annettua tilaa ja siinä vaiheessa, kun taulukko ei enää mahtuisi ruudulle vaan levittäisi näkymää, niin se piilottaa taulukon viimeisen sarakkeen. Samalla se toi nappulan taulukon alkuun, jolla piilotetun sarakkeen tiedon saisi näkyviin rivin alapuolelle.

Tämä kirjasto kuitenkin jouduttiin hylkäämään projektiin loppuvaiheessa, kun responsiivisuutta päästiin kunnolla testaamaan. Se ei ollut yhteensopiva layoutin suhteen. Tähän kokeiltiin toista vastaavaa kirjastoa jQuery:n DataTable, joka pääsääntöisesti olisi tehnyt saman kuin DataTables, mutta sekään ei ilman liiallisia viritelmiä toiminut muun layoutin kanssa.

Lopulta päädyttiin toteuttamaan taulukko HTML:n taulukko ominaisuuksilla, ilman erillistä JavaScript -kirjasto. Responsiivisuus toteutettiin CSS määritelmillä. Taulukolle määritetyn alueen ulkopuolelle menevät taulukon osat piilotetaan, mutta niihin käyttäjä pääsee käsiksi taulukossa silloin näkyvillä vierityspalkeilla.

## 6 Palvelun toteutus

Käytännön toteutus aloitettiin rakentamalla palvelun eri osa-alueista toimivat testiversiot erilliseen testiympäristöön. Ratkaisuna tämä oli ajan ja testaamisen kannalta tehokkain ratkaisu, jolloin pysyttiin keskittymään vain yhden osa-alueen koodin tarkasteluun ja testaamiseen kerrallaan. Ilmanet on kuitenkin laajempi kokonaisuus, jolloin suoraan siellä kehittäessä olisi joutunut ottamaan huomioon paljon muunkin koodin vaikutuksia heti alkuvaiheessa. Osa-alueet koostuivat HTML ja JavaScript -koodista, jotka testiversioiden jälkeen siirretään Ilmanetin PHP -koodiin, joka tuottaa siellä näkyvät sivut.

## 6.1 Kartan rakentamisen

Tässä kappaleessa esitellään karttaosion luominen. Kartan toteuttaminen muodostuu seuraavista vaiheista: peruskartta, reitit, hallinta painikkeet, säätiedot kartalla, koordinaatti muunnokset ja karttakokonaisuus.

### 6.1.1 Kartan perusominaisuudet

Kartan luominen alkoi hakemalla tarvittavat JavaScript kirjastot OpenLayers:in sivustolta ja rakentamalla sille tarpeellinen HTML -sivusto, missä se otettiin käyttöön ja luotiin elementti mihin kartta tulisi. JavaScript tiedostoon tehtiin tarvittavat määritelmät, että saatiin perus kartta tulemaan HTML -sivulle. Tämän jälkeen oli karttamateriaalin vaihtaminen Ilmatieteen laitoksen GeoServer:in tarjoamiin karttamateriaaleihin. Tämän ominaisuuden toteuttamiseen OpenLayer:sin esimerkkisivustolta löytyi ohje, miten uudessa versiossa otetaan WMS-layer karttamateriaali käyttöön. Kyseisen esimerkin avulla saatiin vaihdettua karttamateriaali Ilmatieteen laitoksen omaan karttamateriaaliin. Samalla vaihdettiin kartassa käytetty projektiotukemaan koko maailman tukevaan projektiioon. Siihen valittiin GeoServer:in tukema projektiio EPSG:4326. Karta perusominaisuuksiltaan testattiin ja todettiin toimivaksi.

### 6.1.2 Reitti ominaisuudet

OpenLayers:n esimerkki sivustolta löytyi malli miten reittiominaisuuden voi toteuttaa. Tämän esimerkin pohjalta toteutettiin pohja. Haasteellisempaa oli tässä vaiheessa miettiä miten sen saisi toimimaan useamman reitin kanssa. Reittien kokonaismääräksi oli suunniteltu viisi reittiä. Tässä päädyttiin käyttämään JavaScript ominaisuuksia hyväksi ja jättämään kartta -API:n huoleksi vain yhden reitin käsitteleminen kerrallaan. Kartta -API:lta sai aina ulos reitin koordinaatti tiedon, joka tallennettiin JavaScript objektiin, josta se voidaan tarvittaessa ladata takaisin kartta -API:in. Reitin piirto ja muokkaus ominaisuuden kanssa täytyi miettiä ja määrittää, missä tilanteissa reittiä voisi muokata, jotta näkyvä reitti vastaisi aina sitä mitä dataa kartalla milloinkin on näkyvissä. Ratkaisuna tähän oli, että reittiä voisi muokata vain silloin, kun on valinnut uuden reitin piirrettäväksi tai kun on olemassa olevan reitin kohdalla valinnut muokkaa, muuten reittiä ei pysty muokkaamaan.

### 6.1.3 Kartan hallinta liittymän suunnittelu ja toteuttaminen

Toiminnallisuus karttaan toteutetaan reittien valintaa varten tarvittavilla painikkeille sekä niiden editointia varten tarvittavilla painikkeilla. Reitti painikkeiden suhteen päädyttiin tekemään ne samalla tavalla, kuin Ilmanetin monissa muissakin tuotteissa on toteutettu, eli kartan yläpuolelle niin sanotulla välilehti -mallilla, vaikkei nämä varsinaisia välilehtiä olekaan vaan painikkeita.

Logiikka reittipainikkeiden toimivuuteen osoittautui hieman monimutkaisemmaksi, jotta painikkeet toimisivat kaikissa tapauksissa oikein. Reittipainikkeita päädyttiin toteuttamaan niin, että niitä on maksimissaan viisi kappaletta. Jos reittejä on alle viisi, niin silloin siinä pitää olla painike, josta voi tehdä uuden reitin, mutta jos reittejä on jo viisi, niin silloin siinä ei saa enää olla painiketta uudelle reitille, mutta sen pitää taas ilmestyä, jos yksi reitti poistetaan. Tämä toteutettiin tekemällä logiikan JavaScript:llä, jossa katsotaan aina uuden reitin teon tai vanhan reitin poistamisen jälkeen, että montako reittiä siellä on, että tarvitseeko luoda uusi painike uudelle reitille vai ei. Reitien muokkaus nappulat toteutettiin yhden painikkeen alle ”Muokkaa”, jota painamalla ilmestyy muokkaamiseen tarvittavat lisäpainikkeet (Kuva 1).



Kuva 1: Muokkaa valikon sisältö

### 6.1.4 Sää tietojen esittäminen kartalla

Karttaan sää tiedon esittämistä varten OpenLayers:stä löytyi valmis ominaisuus, millä pystyi piirtämään erilaisia muotoja kartalle ja määrittämään niille värit. Aikaisempia palveluita silmällä pitäen valittiin samankaltainen menetelmä niiden toteuttamiseen eli värjätyt pallot, joilla havainnollistetaan säätä. Tämän ominaisuuden toiminta varmuudeksi toteutettiin tälle oma taso kartalle, mille nämä piirretään, jotta niiden vaihtaminen ja piilottaminen olisi helpompaa, ja että ne eivät sekoittaisi muita kartan elementtejä.

Kuten jo aikaisemmista merenkulkijoille suunnatuista palveluista heille tutuksi tullut tuulennopeus määritettiin vaikuttavaksi tekijäksi kartalle. Näin ollen eri tuulennopeus rajat ja niitä vastaavat värit sai otettua suoraan jo olemassa olevasta palvelusta. Tämä ominaisuus toteutettiin määrittelemällä ennalta värit koodiin, mitä kaikki mahdollisia värejä voisi olla käytös-

sä, josta pallojen piirtologiikka tarkisti dataa piirtäessä jokaisen piirrettävän pisteen, eli datan koordinaatin sekä tuulennopeuden ja niiden perusteella määrittä sille värin ja piirsi koordinaattien osoittamaan paikkaan.

Kartan palloihin lisättiin vielä ominaisuus tuoda lisätietoa paikasta lisäämällä siihen ponnahdusikkunan, kun hiiren vie siihen päälle. Ponnahdusikkunassa näkyy pisteen koordinaatit sekä tuulennopeus ja -suunta. Ponnahdusikkunan tekemiseen löytyi valmis ratkaisu bootstrap kirjastosta. Johon määritettiin vain CSS muotoilut vastaamaan palvelun ulkoasua.

Data palloihin lisättiin myös ominaisuus, jolla taulukon dataa voidaan vaihtaa klikkaamalla niitä. Tämä ominaisuus vain kertoo taulukon tekevällä funktiolle, mitä koordinaatti on painettu, jolloin taulukko funktio voi esittää halutun koordinaation taulukon datasta.

Jotta kartalla esitettävästä datasta olisi enemmän hyötyä tehtiin kartan alapuolelle aikajana, josta kartalla näkyvän datan ajan hetkeä voi vaihtaa. Tähän löytyi valmis ratkaisu jQuery:n - kirjastosta, joka oli jo käytössä Ilmanetissä. Siihen toteutettiin CSS muotoilut, jotta se vastaisi paremmin palvelun ulkoasua, ja lisättiin aikaleimat näkyviin. Aikajanan ajanhetkeä muuttamalla kartalle piirretään aina sitä vastaavat data pallot.

Kartalla näytettävän datan selkeyttämiseksi oli tarpeellista toteuttaa myös legenda. Legendasta käyttäjä pystyy tulkitsemaan, mitä kyseiset kartalla esitettävät pallot tarkoittavat. Jotta mahdollisissa muutoksissa kartalla esitettäviin väreihin, legendan toteutettiin käyttämällä värejä suoraan kartan pallojen piirto koodista ja näin ollen varmistamaan, että ne täsmäävät aina siihen mitä kartalla on piirretty. Legenda muuten toteutui vain laittamalla kartan päälle div elementti ja määrittämään siihen kaikkien käytettyjen värien värikoodit laatikoihin ja ottamalla niitä vastaavat selitteet koodista.

#### 6.1.5 Koordinaatio muunnokset karttaan piirrettäessä

Karttaa toteuttaessa otettiin myös huomioon sen mahdollinen projektio muutos tarpeen tullen. Joten kaikki koordinaatti käsittelyt sisältävät koordinaatti muunnoksen kartan ja datan projektion välillä. Tämä siitä syystä, että eri projektiot käsittelevät koordinaatteja eri tavalla ja datan hakemiseen tarvitsen aina saada koordinaatit EPSG:4326 formaatissa, eli ihmisille tutuimmassa koordinaatti formaatissa.

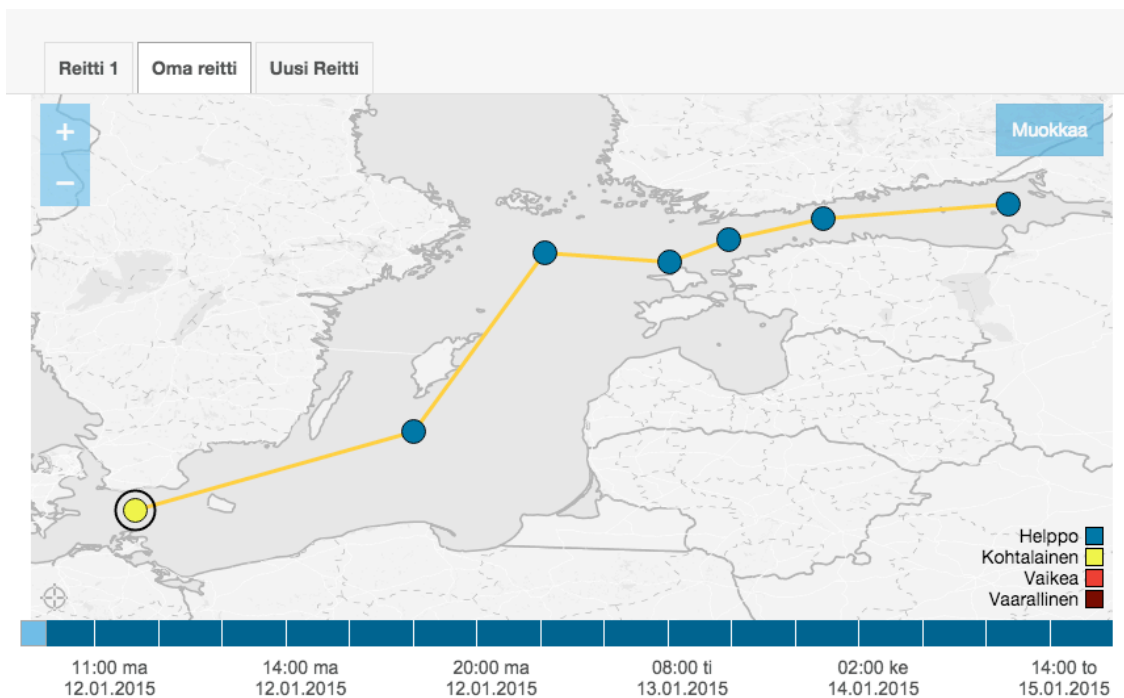
Eli toisin sanoen kartta voi aina käsitellä koordinaattinsa haluamassaan projektiossa, mutta silloin kun niitä on tarve käyttää datan hakemisessa, muunnetaan ne datan tarvitsemaan formaattiin. Datarajapinta palauttaa datan EPSG:4326 formaatissa eli perus koordinaatistossa ja

se tallennetaan palveluun siinä muodossa. Kun dataa tarvitaan esitettäväksi, kartalle tehdään taas muunnos, jotta varmistutaan, että se tulee näkyviin oikealle paikalle kartassa.

#### 6.1.6 Kartta kokonaisuus

Kartta kokonaisuus oli tässä vaiheessa suhteellisen valmis. Yhden palvelua markkinoivan henkilön toiveesta siihen vielä lisättiin tässä vaiheessa paikannusmahdollisuus, sitä tukeville selaimille. Paikannusnappula tulee vain näkyviin silloin, kun selain pystyy sanomaan palveluun tullessa, että tukee kyseistä ominaisuutta, muuten se on piilotettu. Tämä sen takia ettei sen takia tule turhia väärinkäsityksiä, ettei se toimisi ja minkä takia. Ja se myös toteutettiin ei niin näkyvänä ja huomiota herättävänä elementtinä, koska sen toimivuus ja tarkkuus eivät HTML maailmassa ole äärimmäisen luotettava ja sen varaan liikkussa ei voi luottaa. Joten se on vain suuntaa antava lisä palvelussa.

Kokonaisuudessaan kartta oli nyt valmis (Kuva 2), missä voi nähdä miten eri kartan elementit on sijoiteltu. Taulukossa esitettävän pisteen datan selkeyttämiseksi kartan valittuun pisteeseen luotiin vielä lisä rengas ympäröimään sitä. Jotta se havainnollistaa paremmin käyttäjälle, että nyt taulukossa on kyseisen pisteen data näkyvillä.



Kuva 2: Palvelun kartta näkymä

## 6.2 Taulukon rakentaminen

Taulukon rakentaminen aloitettiin tarvittavan JavaScript -kirjaston hankkimisella. Tämän jälkeen tehtiin tarvittavat HTML elementit, jotta sen pystyisi alustamaan JavaScript:ssä. Alkuvaiheessa syötettiin siihen vain testidataa, jotta pystyin kokeilemaan sen toimivuutta. Ja sen toimintavarmuus ei täyttänyt kaikkia odotuksia. Välillä se jätti tekemättä asioita, joita sen olisi ollut tarkoitus tehdä, kun pienentämättä taulukon aluetta kun käytettävissä oleva tila muuttuu.

Ratkaisuna tähän päädyttiin tekemään taulukko täysin ilman JavaScript -kirjastoa. Joten se toteutettiin rakentamalla HTML:ään perus pohjan taulukolle, ja määrittämään sille CSS tyyliin sitä sisällä pitävälle elementille määritelmät koon suhteen. Näillä asetuksilla saatiin tapahtumaan, se mitä taulukolta haluttiinkin eli responsiivisen toiminnallisuus.

Koska taulukossa vaihtuu data ei sitä voinut kokonaan luoda HTML:ssä, joten JavaScript:llä tehtiin funktio, joka tekee taulukkoon elementtejä rivi riviltä käyttäen dataa, mitä palveluun on haettu. Tyyli taulukkoon saatiin ilmanetissä jos olemassa olevasta taulukko tuotteesta.

Kokonaisuutena (Kuva 3) on HTML taulukko, mutta se rakentuu tilauksen asetusten mukaan ja data sisältöä voidaan muuttaa nappia painamalla. Taulukon ylä- ja alaotsikot luodaan PHP:lla, joka luo sivun pohjan, missä saadaan tehtyä siihen kukin asiakkaan haluamat parametrit.

13.4807589, 55.0289199

Alka	Tuuli m/s	Tuulen-	suunta	Puuska m/s	Aallonkorkeus	Lämpötila	Kosteus %	Sää	P (hPa)
10:00 ma 12.01.2015	13.3	SW	230	19.1	2.1	5.4	89.7		1002.9
11:00 ma 12.01.2015	14	SW	230	20.6	2.3	5.5	88.7		1002.7
12:00 ma 12.01.2015	13.3	SW	230	20.7	2.5	5.7	87.7		1002.3
13:00 ma 12.01.2015	13.6	SW	230	19.9	2.7	5.7	89.1		1001.9
14:00 ma 12.01.2015	14	SW	230	20.3	2.9	5.9	88		1001.5
15:00 ma 12.01.2015	14	SW	230	20.2	2.8	6	87.9		1001.2
17:00 ma 12.01.2015	13.9	SW	230	19.8	2.6	6.2	87.7		1000.4
20:00 ma 12.01.2015	13.9	SW	230	19.4	2.5	6.4	89.6		1000.3
23:00 ma 12.01.2015	13.1	WSW	240	18.4	2.2	6.9	88.2		1001.2
02:00 ti 13.01.2015	10.9	WSW	250	15.2	1.7	6.7	86.5		1003.2
08:00 ti 13.01.2015	6.7	SW	220	9	1.1	5.2	82.1		1002.4
14:00 ti 13.01.2015	13.5	SSW	210	18.4	1.4	6.7	89		996.6
20:00 ti 13.01.2015	10.8	WSW	240	15.2	1.7	5.6	86.4		998.7
02:00 ke 14.01.2015	10.1	SSW	210	13.4	1.6	4.9	87.4		1000.2
14:00 ke 14.01.2015	13.3	WSW	240	18.8	-	4.1	78.8		1004
02:00 to 15.01.2015	11.6	WSW	240	15.5	-	3.7	82.5		1007.3
14:00 to 15.01.2015	13.5	S	180	19.1	-	5.4	82.3		996.3
02:00 pe 16.01.2015	9.6	SW	230	13.2	-	6	91.5		992.8
Alka	Tuuli m/s	Tuulen-	suunta	Puuska m/s	Aallonkorkeus	Lämpötila	Kosteus %	Sää	P (hPa)

Kuva 3: Palvelun taulukko näkymä

### 6.3 Kokonaisuuden rakentaminen

Palvelun kokonaisuuden rakentaminen lähti liikkeelle määrittelemällä Ilmanettiin pohja uutta tuotetta palvelua varten. Tämä sisältää määritelmät, mitä kaikkea palveluun voidaan muokata tilauskohtaisesti, kun se otetaan asiakkaalle käyttöön. Tähän kuului myös palvelun tilauspohjan rakentaminen, missä kyseisiä määriytyksiä voidaan muokata tilauskohtaisesti.

#### 6.3.1 Ilmanet

Ensin määritettiin palvelulle perusasetukset, joita Ilmanetin palvelut tarvitsevat. Nämä asetukset saatiin kopioitua valmiista pohjasta uusien palveluiden luomista varten. Tämän jälkeen Ilmanettiin tuotiin testiympäristössä toteutetut palvelun osat ja määritettiin ne Ilmanetin PHP-koodiin ladattavaksi, kun palveluun tullaan. Näihin sisältyi JavaScript ja CSS tiedostot. HTML koodit täytyi sisällyttää palvelun rakentavaan PHP koodiin, jolloin niitä voidaan muokata tilauskohtaisesti tarpeen mukaan.

Näiden perus asetusten jälkeen palvelun pohja Ilmanetissä oli valmiina, jolloin pystyttiin aloittamaan palvelun osa-alueiden integroiminen palveluun.

#### 6.3.2 Kartan integroiminen

Kartan siirtäminen Ilmanet ympäristöön toteutui vain siirtämällä koodi Ilmanet puolelle. Ja tarvittavat tiedostot yhdistettiin Ilmanetin peruskoodiin, mikä lataa ne palveluun tullessa. Kaikki kartan ominaisuudet testattiin tässä vaiheessa, ja ne toimivat samalla tavalla myös Ilmanet ympäristössä.

#### 6.3.3 Taulukon integroiminen

Taulukon kohdalla toimittiin melkein samalla tavalla kuin kartankin kanssa, mutta tässä tapauksessa JavaScript koodi siirrettiin samaan tiedoston, missä kartankin toiminnallisuus oli. Tämä tehtiin sen takia, että taulukon osalta tarpeellista koodia oli vain yhden funktion verran, ja olisi ollut tarpeetonta pitää sitä erillisessä JavaScript tiedostossa.

#### 6.3.4 Datahaut

Datahauille on Ilmanetissä olemassa peruspohja, mitä Ilmanetin palvelua luodessa voi helposti muuttaa, jotta voi saada dataa haluamallaan tavalla. Tähän palveluun datahaku tavaksi oli valittu JSON-formaatissa hakeminen, koska sitä on helppo käsitellä JavaScript:ssä.



Nykyinen datahaku menetelmä ei kuitenkaan ongelmitta taittunut palvelussa tarvittaviin määritelmiin. Kun dataa haki monesta eri lähteestä, ei nykyinen datahaku menetelmä osannut yhdistää dataa yhdeksi, koska merellisillä alueilla ei ole minkäänlaisia id:tä. Nykyinen datahaku menetelmä vaati, että haetuilla paikoilla olisi määritetty id:t, jotta niitä voisi yhdistää. Tähän ratkaisuna tehtiin datan käsittely menetelmään uusi ominaisuus, jotta se osaa yhdistää datat tarpeen mukaan myös koordinaattien perusteella. Tämä ominaisuus täytyi huolella testata, ettei se rikkoisi mitään muuta datanhaku menetelmää. Ominaisuuden toteuttamiseen oli kuitenkin valittu varmin keino muiden datahakujen eheyden varmistamisen kannalta. Datahaku menetelmään täytyi lisätä uusi muuttuja, jotta saa datan palveluun tarvitsemalla tavalla, tällöin myös varmistui muiden palveluiden toimivuus, mutta ne testattiin silti vielä erikseen.

Ilmanet palvelun puolella tarvitsi vielä määritellä muutama asia, jotta se pystyy kutsumaan datahakua asiakkaan tilaukseen määritetyillä parametreilla ja muuttamaan sen JSON-formaattiin. Datan alkuun tehtiin vielä lisämäärittely johon lisättiin tieto datan sisältämistä aika-askelista. Tämä toteutettiin PHP koodin puolella, koska silloin se tarvitsi tehdä vain kertaalleen, eikä useampaan kertaan, kuten sen olisi joutunut JavaScript:in puolella tekemään. Se myös hieman nopeutti palvelun toimivuutta käyttäjän näkökulmasta, kun ei aina tarvitse käydä koko dataa läpi, jotta voidaan määrittää aikajanalla näkyvät ajanhetket.

Palvelun datahaku kutsu yhdistettiin kartan toiminnallisuuteen seuraavasti. Kun kartassa tallentaa uuden reitin tai kun tallennettu reitti ladataan näkyville, niin sen jälkeen kutsutaan datahakua kartasta saaduilla koordinaateilla.

#### 6.3.5 Kokonaisuuden toiminnan yhdistäminen

Tässä vaiheessa kartta ja taulukko molemmat olivat täysin toimivia, mutta ne eivät vielä toimineet yhteen. Niiden saamiseksi toimimaan yhteen piti kartan toiminnallisuuksiin lisätä funktio, mikä kutsuu taulukon rakentamista. Kartta osion koodista kutsutaan taulukkoa piirtävää funktiota, kun kartalle on saatu piirrettyä data ja valittua siitä ensimmäinen piste näytettäväksi, jolloin tiedetään minkä pisteen data pitää saada myös taulukkoon näkyville.

Palvelusta puuttui kuitenkin vielä yksi tärkeä toiminnallisuus, joka oli reittitietojen tallentaminen, jotta ne olisivat käytössä myös seuraavalla kerralla, kun tulee palveluun. Tämä toteutettiin välittämällä tallennusvaiheessa kartan antamat koordinaatit ja reitille valitun nimen JSON-formaatissa Ilmanetille. Ilmanetin puoleisessa PHP koodissa käytettiin valmista funktiota, jolla pystyttiin tallentamaan se tietokantaan, sidottuna asiakkaan tilaukseen. Jotta nämä saadaan käyttöön palveluun tullessa, täytyi myös tehdä niille haku palvelua alustettaessa, tämä toteutettiin tekemällä valmiilla funktiolla haku Ilmanetin tilauksia koskevaan tietokantaan ja hakemalla sieltä kaikki asiakkaan reittitiedot ja sijoittamalla ne palvelun ladatessa

HTML:ään JavaScript muodossa, jotta palvelun varsinainen JavaScript tiedosto osaisi ottaa ne käyttöön. Tämän jälkeen piti vielä palvelun varsinaiseen JavaScript:iin tehdä muutos, että jos se löytää määrittämisen aikaisemmin tallennetuista reiteistä, niin se lataa ne muistiin palveluun ja tekee tarvittavat painikkeet kyseisiä reittejä varten ja aktivoi ensimmäisen reitin näkyviin palveluun.

Lisäominaisuutena havainnollistamaan valittua ajanhetkeä päädyttiin tekemään palveluun aktiivisen ajanhetken korostus taulukkoon. Tähän päädyttiin, koska aikajanaan ei millään mahtunut jokaisen ajanhetken kohdalle aikaleimaa näkyviin, varsinkin kun palvelun olisi tarkoitus olla responsiivinen, jolloin se voi olla todella kapea. Tämä toteutui JavaScriptillä lisäämällä taulukkoon CSS tyylittely silloin, kun aikajanalla on valittuna kyseinen ajankohta ja muuten poistamalla se.

#### 6.4 Responsiivisuus

Responsiivisuudella tarkoitetaan mukautuvuutta. Responsiivisesti toteutettu sivusto tunnistaa käytettävän laitteen ja mukauttaa sisällön, ulkoasun ja erilaiset toiminnallisuudet automaattisesti käytettävän laitteen mukaan, joten erilaiset mobiilisivustot ja saman sisällön päivittämiset useaan paikkaan on historiaa. (Responsiivisuus info 2015)

Mobiili laitteiden yleistyessä jokapäiväisessä käytössä oli erittäin tärkeätä, että palvelu toimi myös niissä moitteettomasti eli toimisi responsiivisesti. Palvelun responsiivisuuden tärkeyteen vaikutti myös se tekijä, että palvelun toteuttamisajanhetkellä Ilmanetti oli siirtymässä responsiiviseen ulkoasuun, joten kaikkien uusien palveluiden tulisi tukea myös sitä.

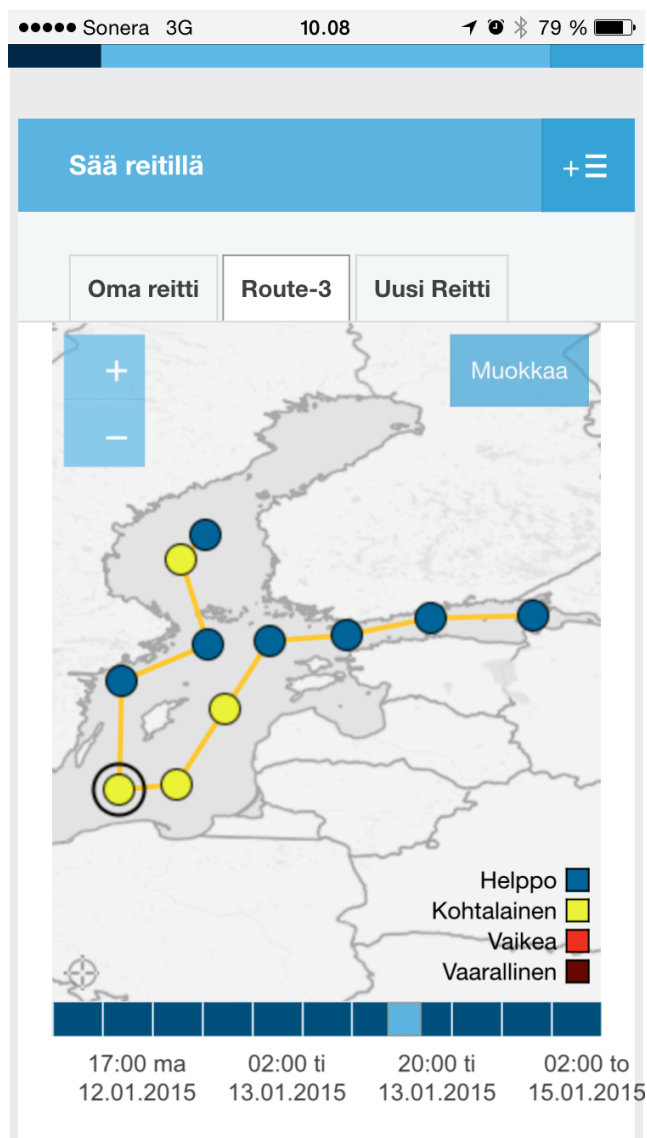
Ilkka Olander:in mukaan responsiivinen toteutus on kannattavampi kuin muut mobiililaitteita varten olevat ratkaisut ja listaa seitsemän syytä sille.

1. Yksi sivusto joka toimii kaikilla laitteilla
2. Responsiivinen sivusto suunnittelu on muita vaihtoehtoja nopeampaa ja edullisempaa
3. Yhtenäinen visuaalinen ilme
4. Hakukoneystävällisyys
5. Verkkoanalytiikka
6. Helppo ylläpitää ja päivittää
7. Responsiivisuus on tulevaisuutta (Olander 2014)

### 6.4.1 Kartan responsiivisuus

Responsiivisuuden testaamisen aloitettiin kartan toiminnallisuudesta, koska se olisi kuitenkin tärkein osa saada toimimaan. Oli tiedostettu, että kartta -API:n pitäisi tukea valmiiksi responsiivisuutta. Ja alun testivaiheessa oli sitä jos testattu ja todettu toimivaksi. Ilmanet ympäristössäkin kartta toimi responsiivisena moitteettomasti (Kuva 4).

Kuitenkin pieniä muutoksia kartan painikkeisiin jouduttiin tekemään, jotta ne toimisivat paremmin myös pienemmillä näytöillä. Nämä muutokset saatiin tehtyä pelkillä CSS määrittämisillä.



Kuva 4: Mobiili näkymä palvelun kartasta

## 6.4.2 Taulukon responsiivisuus

Taulukon responsiiviseksi saaminen osoittautui haastavammaksi kuin ennalta oli arvioitu, koska siihen ei valmiit responsiiviset JavaScript kirjastot kyenneet Ilmanet ympäristössä. Ratkaisuna tähän ongelmaan tehtiin taulukon CSS määrittäisiin muutoksen, jotta se sallii taulukon menevän sille määritetyn alueen ulkopuolelle, mutta niin että ulkopuolelle menevä alue ei ole näkyvillä. Eli koko taulukko on olemassa, mutta se osa mikä ei ruudulla näy on saatavilla, kun sen kohdalla siirtyy oikealle tai vasemmalle, joko hiirellä käyttäen virityspalkkia tai mobiilikäytössä sormella vetämällä sivusuuntaan (Kuva 5).

17.5293031, 55.6447665

Aika	Tuuli m/s	Tuulen- suunta	Puuska m/s	Aa
14:00 ma 12.01.2015	14.6	WSW	240	21.4
17:00 ma 12.01.2015	14.3	WSW	240	21
20:00 ma 12.01.2015	14.8	SW	230	19.6
23:00 ma 12.01.2015	15.6	WSW	240	21.2
02:00 ti 13.01.2015	14.2	WSW	250	21.8
08:00 ti 13.01.2015	10.2	WSW	250	13.8
14:00 ti 13.01.2015	10.6	SSW	192.8	15.1
20:00 ti 13.01.2015	14.9	SW	230	21.3
02:00 ke	9.3	WSW	240	12.6

Alonkorkeus	Lämpötila	Kosteus %	Sää	P (hPa)
2.9	5.6	90.8		999.4
3.9	5.8	89.6		998.9
3.9	6.2	89.3		998.6
4	6.5	88.7		998.3
3.9	6.9	87.3		999.7
2.1	5.8	79.7		1002.5
1.9	5.5	93.2		998.9
3.1	6.5	87.8		995.7
2.4	5.5	88.3		1000.4

Kuva 5: Mobiili näkymä taulukosta

## 6.5 Tilauksen muokattavuus

Ilmanet asiakas palvelukanavassa on erillinen paikka, missä tilauksia voidaan tehdä ja muuttaa niiden asetuksia sen mukaan mitä mikäkin palvelua on määrittänyt niille muutettaviksi. Näkymä (Kuva 6) palvelun muokkaus näkymästä. Kaikki nämä asetukset olivat jo valmiiksi tehtyjä Ilmanetin tilausten hallintaan, ne piti vain määrittää, että mitä niistä halutaan ottaa käyttöön kyseisessä palvelussa.

The screenshot shows a configuration interface for a weather service. At the top left, there is a blue button labeled 'Tallenna asetukset'. Below it, the 'Aikaleiman muoto' (Time format) is set to 'TT:MM päivä PP.KK.VVVV'. The 'Ennusteen ajoitus' (Forecast timing) section shows 'kehys / väli / siirto' (frame / interval / shift) with '1 tunti' (1 hour) selected for the interval. There is a '+ Näytä lisäasetukset' (Show additional settings) link and a 'Mukauta tilauksen asetuksia' (Customize service settings) checkbox. A search bar contains 'windcom'. Below the search bar are two columns of service parameters. The left column lists 'WindCompass16', 'WindCompass32', and 'WindCompass8'. The right column lists 'WindSpeedMS' (highlighted in yellow), 'WindCompass16', 'WindDirection', 'Maksimipuuska 10min', 'SigWaveHeight', 'Temperature', 'Humidity', 'WeatherSymbol3', and 'Pressure'. To the right of these columns are several other settings: 'Tietomalli' (Data model) set to 'Oletus' (Default), 'Parametrin nimi' (Parameter name) set to '(Tuuli m/s)', 'Taso' (Level), 'Yksikkömuunnos' (Unit conversion), and 'Symboolikokoelma' (Symbol set).

Kuva 6: Tilauksen muokkausnäky

Ensimmäisenä muokattavana asiana on aikaleiman muoto, mitä palvelussa käytetään. Näitä vaihtoehtoja on asiakkaiden erilaisista mieltymyksistä kertynyt useampia. Sieltä löytyy vetovalikon takaa niin yleisimmin käytetyt aikaleima muodot kuin myös hieman harvemmin käytetyt.

Toisena oli palvelun data hakuja koskeva aikamääritelmien asettaminen. Siinä pystyi valitsemaan aika-askleen, mitä käytetään ja kuinka pitkään dataa esitetään. Myös ennusteen alkamista pystyi siirtämään, jos vaikka haluaisi ensimmäisen ajan hetken olevankin useamman tunnin kuluttua. Aikajaksoja pystyy asettamaan useampia, jolloin esim. on mahdollista tehdä tilaus, jossa alkuun on tiheämmällä aikavälillä ennusteita ja loppua kohden harvenevilla väleillä.

Lisäasetuksien alta löytyy sivun päivittämistiheyden määrittäminen, eli kuinka usein data päivitetään, jos palvelun jättää auki ruudulle pitkäksi aikaa.

Loput asetuksen ovat tilattavien sääparametrien asettelua. Siinä pystyy hakemaan kaikkia parametreja, joita Ilmanettiin on asetettu ja vetämään ne oikeanpuoleiseen laatikkoon (Kuva 5). Oikeasta laatikosta klikkaamalla parametria tulee vielä sivun oikeaan reunaan asetukset parametrikohteisesti. Parametrikohteisissa asetuksissa voidaan määrittää parametrissa käytettävä data lähde eli tietomalli. Parametrin nimen voi myös vaihtaa, koska Ilmanetissä on pääsääntöisesti käytössä englanninkieliset versiot parametrinimistä. Tasolla tarkoitetaan painepintadatoja, joissa samaan pisteeseen löytyy useampia korkeuksia eri painepinta tasoilta. Yksikkö muunnoksia voidaan toteuttaa tietyille parametreille, jos asiakas on tottunut käyttämään jotain tiettyä dataa tietyssä formaatissa. Ja viimeisenä tuuli ja sää symboleita varten oleva symbolikokoelma, jolla saadaan numeromuotoinen data vaihdettua sitä vastaavaksi kuvaksi.

## 7 Sää reitillä -palvelu

Tässä opinnäytetyössä kehitettiin Sää reitillä -palvelu. Palvelussa käyttäjä voi piirtää kartelle viivoilla reittejä, jolle hän haluaa saada säätietoja. Reittejä pystyy tallentamaan muistiin ja muokkaamaan jälkikäteen, jottei aina palveluun tullessa tarvitse määrittää reittiä uudestaan. Tällä ominaisuudella palvelusta saatiin helppokäyttöisempi loppukäyttäjille.

Valitulle reitille haetaan asiakaskohteisesti ennalta määritettyjä parametreja eli säätietoja näkyville. Kartalle tulee värjättyjä palloja näkyviin kuvastamaan tuulen nopeutta valitun reitin pisteillä. Kartan pallojen esittämää ajanhetkeä pystyy muuttamaan kartan alapuolella olevalla aikajanalla. Lisänä kartan pallot antavat vielä hieman lisäinformaatiota pisteestä viemällä hiiren niiden päälle, jolloin ponnahdusikkunaan tulee näkyviin lisätietoja pisteestä (koordinaatit, tuulensuunta ja tuulennopeus).

Kartan alapuolelle muodostuu myös taulukko, missä näytetään kaikki asiakkaalle valitut sääparametrit. Taulukon yläpuolella näkyy koordinaatit, minkä pisteen dataa kulloinkin esitetään taulukossa ja jotta käyttäjä saisi paremman käsityksen mitä pistettä taulukossa esitetään, korostetaan kartalla valittua pistettä esittämällä sen ympärillä musta rinkula. Taulukossa esitettävää pistettä pystytään vaihtaman klikkaamalla kartalta toista pistettä, jolloin taulukko muodostuu uudestaan sisältäen kyseisen pisteen sää tiedot.

Palvelussa näytettävät sääparametrit voidaan määrittää asiakaskohteisesti heidän tilausta tehdessään, jolloin saadaan paremmin asiakkaan tarpeita täyttävä palvelu aikaiseksi. Myös aikajaksoja ja aikaleimojen muotoja voidaan muokata asiakkaan tarpeen mukaisesti.

## 8 Loppuarviointi

Tämän toiminnallisen opinnäytetyön tavoitteena oli luoda Sää reitillä -palvelu Ilmatieteen laitoksen Ilmanet asiakas palvelukanavaan. Opinnäytetyön tavoite saavutettiin ja palvelu saatiin valmiiksi ennalta sovituin ominaisuuksin määräaikaan mennessä. Projekti aikataulu oli pääasiassa hyvin laadittu, ainoastaan responsiivisen taulukon toiminnallisuuden toteuttaminen vei budjetoitua enemmän aikaa, mutta silti pysyttiin määräajassa.

Responsiivinen toiminnallisuus toteutui määritelmien mukaisesti lopullisessa palvelussa, niin kartaan kuin taulukonkin suhteen. Vaikka taulukon responsiivinen toiminnallisuus oli hieman odotettua työläämpi saada toimivaksi.

Työ toteutettiin Ilmatieteen laitokselle sisäisenä projektina. Projektin alussa käytettiin hyväksi aikaisemmin ulkopuolisella firmalla teettämää asiakas kyselyä ja sen tuloksia, jolla saatiin pohja palvelulle. Määritykset olivat, että se toimisi Ilmatieteen laitoksen Ilmanet palvelukanavassa.

Opinnäytetyön teoriaosuudessa keskityttiin sivuston toteuttamisen vaiheisiin ja responsiivisen ratkaisun tärkeyteen ja hyötyihin. Toteutus menetelmissä käytiin läpi ratkaisuja ja toteutus tapojen valintoja palvelun luomisessa.

Palvelusta on tullut positiivista kommentteja potentiaalisilta asiakkailta, jota saattavat tulevaisuudessa haluta tilata kyseisen palvelun. Asiakkailta on tullut myös useita hyviä jatkokehitysideoita palvelulle, miten siitä tehdä vielä paremmin asiakkaan tarpeita palveleva esimerkiksi lisäämällä kartalle lisää dataa.

## Lähteet

Ilmatieteen laitos. 2015. Ilmanet - Parhaat sääennusteet reaaliajassa. Viitattu 18.1.2015

<http://ilmatieteenlaitos.fi/ilmanet>

Cenno. 2014. Mikä on API ja miksi se on SaaS ohjelmistossa niin tärkeä. Viitattu 21.1.2015

<http://blogi.cennoapp.com/2014/05/mika-on-api-ja-miksi-se-on-saas.html>

w3school. 2015. HTML Introduction. Viitattu 21.1.2015

[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

w3school. 2015. JavaScript Tutorial. Viitattu 21.1.2015

<http://www.w3schools.com/js/>

OpenLayers. 2008. What is OpenLayers? Viitattu 21.1.2015

<http://docs.openlayers.org/>

web-opas. 2015. Mikä on CSS? Viitattu 21.1.2015

[http://www.webopas.net/mika\\_css.html](http://www.webopas.net/mika_css.html)

Responsiivisuus info. 2015. Mitä on repsonsiivisuus? Viitattu 18.1.2015.

<http://www.responsiivisuus.info/responsiivisuus.html>

GeoServer. 2014. GeoServer. Viitattu 21.1.2015

<http://geoserver.org>

OpenGeo. 2015. Web Map Service (WMS). Viitattu 21.1.2015

<http://girona-geoserver-workshop.readthedocs.org/en/latest/overview/wms.html>

w3school. 2015. JSON Tutorial. Viitattu 21.1.2015

<http://www.w3schools.com/json/>

PHP. 2015. PHP: Hypertext Preprocessor. Viitattu 21.1.2015

<http://php.net>

Ilmatieteen laitos. 2015. Ilmatieteen laitos. Viitattu 10.1.2015

<http://ilmatieteenlaitos.fi/ilmatieteen-laitos>

Olander, I. 2014. Responsiiviset sivut - verkon minimistandardi 2014. Viitattu 18.1.2015.

<http://sometek.fi/responsiiviset-verkkosivut-minimistandardi-2014/>



## Kuvat

Kuva 1: Muokkaa valikon sisältö .....	11
Kuva 2: Palvelun kartta näkymä .....	13
Kuva 3: Palvelun taulukko näkymä .....	15
Kuva 4: Mobiili näkymä palvelun kartasta .....	19
Kuva 5: Mobiili näkymä taulukosta .....	20
Kuva 6: Tilauksen muokkausnäkymä .....	21