

## **eNolla -sovellus**

Projektin kehitysvaiheen seurantatyökalu



Ammattikorkeakoulututkinnon opinnäytetyö

Riihimäki, Tietotekniikan koulutusohjelma

syksy, 2017

Santeri Tiilikainen

Tietotekniikan koulutusohjelma  
Riihimäki

---

<b>Tekijä</b>	Santeri Tiilikainen	<b>Vuosi</b> 2017
<b>Työn nimi</b>	eNolla -sovellus: Projektin kehitysvaiheen seurantatyökalu	
<b>Työn ohjaaja/t</b>	Petri Kuittinen	

---

## TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli suunnitella ja kehittää verkkoselainpohjainen projektin kehitysvaiheen tavoitteiden seurantatyökalu. Työn toimeksiantajana toimi hämeenlinnalainen ohjelmistoyritys Flowmedik Oy, joka on erikoistunut visuaalisesti näyttäviin ja käyttäjäystävällisiin SaaS-palveluina toteutettuihin ERP-mittareihin.

Tuotteen idea syntyi tarpeesta saada tapa seurata luvattujen tavoitteiden kehittymistä sairaalarakentamisen yhteydessä jo projektin kehitysvaiheessa. Tässä työssä käydään läpi tuotteen tekninen toteutus, kehitykseen valitut työkalut, vaihtoehtoisia työkaluja sekä tuotteen näkymät ja ominaisuudet.

Tuloksena syntyi käyttökelpoinen tuote, eNolla, joka kertoo käyttäjälle mitattavalle kohteelle asetettujen tavoitteiden toteutumisen nykyhetkestä ja historiasta. Tuotteen varsinaisista käyttäjistä voidaan mainita esimerkireferenssinä Pohjois-Karjalan sosiaali- ja terveyspalvelujen kuntayhtymä, jossa eNollaa tullaan käyttämään sote -toiminnan mittaamiseen ja kehittämiseen tavoitepohjaisesti.

**Avainsanat** ohjelmistokehitys, ohjelmointi, projektinhallinta, SaaS, web-kehitys

**Sivut** 36 sivua, joista liitteitä 2 sivua

Information Technology  
Riihimäki

---

<b>Author</b>	Santeri Tiilikainen	<b>Year</b> 2017
<b>Subject</b>	eNolla -application: Monitoring tool for the developmental phase in the project	
<b>Supervisors</b>	Petri Kuittinen	

---

ABSTRACT

The purpose of this project was to design and implement a web browser-based application for monitoring the set goals of a project's developmental phase. The commissioner of the project was a software company Flowmedik Oy, which is Hämeenlinna based and specializes in visually appealing and user-friendly SaaS ERP meters.

The idea for the product arose from the need to have a method for monitoring the development of the promised goals during a project's developmental phase whilst building a hospital. This thesis covers the technical execution, the tools chosen for the task, some alternative tools and the views and features of the product.

The result of the project was a usable product, eNolla, which tells the user about the current and historical state of how the monitored subject has met the set goals. One of the actual reference customers of eNolla is the municipal federation of North Karelia's social and health services where it will be used to measure and improve operations based on their goals.

**Keywords** programming, project management, SaaS, software development, web development

**Pages** 36 pages including appendices 2 pages

## TERMEJÄ JA LYHENTEITÄ

Suomi	English	Selitys
Avoin lähdekoodi	Open source	Ohjelmien kehitysmenetelmä, joka sallii ohjelman levittämisen, käyttämisen ja muokkaamisen vapaasti.
CDN, hajautettu sisällönjakelu	Content Delivery Network	Tapa jakaa palvelinten sisältöä maailmanlaajuisesti toiminnan nopeuttamiseksi.
Dynaaminen verkkosivu	Dynamic web page	Verkkosivu, joka luodaan palvelimella selaimen pyynnöstä esimerkiksi tietokannassa olevien tietojen perusteella.
ERP, toiminnanohjausjärjestelmä	Enterprise Resource Planning	Kokoelma integroituja toimintoja, joilla pyritään tehostamaan yritysten tehokkuutta.
IDE, ohjelmointiympäristö	Integrated development environment	Ohjelmisto, joka tarjoaa työkaluja ohjelmistokehitykseen.
Kirjasto	Library	Kokoelma resursseja, kuten aliohjelmiä, joka mahdollistaa niiden käytön ilman, että niitä tarvitsee kirjoittaa aina uudelleen.
MVC	Model-view-controller	Ohjelmistoarkkitehtuurityyli, jota käytetään yleensä graafisten käyttöliittymien kehityksessä.
Ohjelmistokehys	Software framework	Ohjelmistokehityksen nopeuttamiseksi tarkoitettu työkalu.
Ohjelmointiparadigma	Programming paradigm	Malli ohjelmointikielten luokitteluun niiden ominaisuuksien mukaan.
Refaktorointi	Code refactoring	Lähdekoodin rakenteen parantaminen ilman, että sen toiminnallisuus muuttuu.
SaaS	Software as a Service	Yleensä verkkoselaimella käytettävä palveluntarjoajan ylläpitämä ohjelmisto.
Staattinen verkkosivu	Static web page	Verkkosivu, joka ei muutu ilman, että sitä muokataan palvelimella.

## SISÄLLYS

1	JOHDANTO.....	1
2	TIETOPERUSTA JA TYÖKALUT .....	1
2.1	Käytettyjä työkaluja.....	2
2.1.1	HTML.....	2
2.1.2	CSS .....	3
2.1.3	JavaScript.....	4
2.1.4	PHP .....	4
2.1.5	jQuery .....	5
2.1.6	Bootstrap .....	5
2.1.7	Google Charts .....	6
2.1.8	MySQL.....	7
2.1.9	NetBeans .....	7
2.2	Vaihtoehtoisia työkaluja.....	8
2.2.1	Python.....	8
2.2.2	Laravel .....	10
2.2.3	ReactJS.....	10
2.2.4	MariaDB.....	11
2.2.5	Eclipse .....	12
3	TUOTTEEN SUUNNITTELU .....	12
4	OMINAISUUDET JA NÄKYMÄT.....	13
4.1	Case: Polikäynnit .....	13
4.2	Päänäkymä .....	13
4.3	Infokenttä.....	14
4.3.1	Kehitysaika.....	15
4.3.2	Tavoite .....	15
4.3.3	Reagointisuunta.....	16
4.3.4	Toleranssi.....	16
4.3.5	Poikkeamaluku .....	17
4.3.6	Seuranta-aikaväli .....	18
4.3.7	Viimeisin päivitys .....	18
4.4	Mittarivalikko .....	19
4.5	Päänäkymän luku .....	19
4.6	Tilastot.....	20
4.6.1	Kaikki merkinnät -kaavio .....	21
4.6.2	Keskiarvokaavio .....	22
4.6.3	Aikajanakaavio.....	22
4.7	Lisäasetukset .....	23
4.8	Upotenäkymä .....	24
4.9	Mittarinhallintatyökalut .....	25
4.9.1	Uuden mittarin luonti.....	26
4.9.2	Vanhan mittarin muokkaus .....	27
4.10	Syöttötyökalu .....	30

5	POHDINTA JA YHTEENVETO.....	31
	LÄHTEET.....	33
	Liitteet	
	Liite 1	eNolla, etusivu
	Liite 2	eNolla, tilastonäkymä

## 1 JOHDANTO

Tämän opinnäytetyön aiheena on suunnitella ja kehittää eNolla-niminen web-sovellus hämeenlinnalaiselle Flowmedik Oy -nimiselle ohjelmistoyritykselle. Yritys on erikoistunut sosiaalitoimen ja terveydenhuollon toiminnanohjausmittareihin. Yrityksen tuotteet ovat yleisesti nimeltään mallia ”eTuote”.

Tavoitteena on kehittää toiminnanohjausmittari, jonka tarkoituksena on kertoa käyttäjälle kehitysprojektille asetettujen tavoitteiden toteutumisesta jo kehitysvaiheen aikana. Tavoitteet voivat olla mitä vain yksiköissä mitattavaa, kuten tehtyjä suoritteita, euromääriä tai tehtyä työaikaa. Päänäkymän tulisi olla yksinkertainen, visuaalinen mittari, josta näkee nopeasti vilkaisemalla missä suhteessa tavoitteeseen ollaan. Mittarin kautta tulisi myös päästä käsiksi mittarin historiatietoihin, joista näkisi kehityksen suunnan sekä projektin aikajanan. Mittari on osa suurempaa projektinhallinta -ohjelmistokokonaisuutta, joka koostuu yhteensä neljästä eri sovelluksesta. Se on kuitenkin toimiva tuote myös erillään käytettävänä. Mittarin nimi, eNolla, perustuu siihen, että mittarin päänäkymän idea on kertoa käyttäjälle yksinkertaisella, visuaalisella tavalla kuinka paljon tarkasteluhetkellä on heittoa tavoitteesta eli nollasta.

Oma työni keskittyy mittarin suunnitteluun, itse sovelluksen ohjelmointiin sekä sen jatkokehitykseen sovelluksen käytöstä syntyneiden kehitysideoiden ja saatujen tulosten mukaan. Tämä opinnäytetyö keskittyy pääasiassa tuotteen tekniseen toteutukseen ja ominaisuuksiin. Käydään läpi eNollan kehitykseen käytetyt menetelmät ja työkalut sekä tuotteen näkymät ja niiden käyttäjälle tarjoama informaatioarvo. Esitellään myös joitain muita suosituimpia työkaluja ja menetelmiä, joilla tuotteen olisi mahdollisesti voinut toteuttaa tai joita olisi voinut käyttää muuten hyödyksi.

## 2 TIETOPERUSTA JA TYÖKALUT

Tämä luku käsittelee eNollan kehitykseen käytettyjä työkaluja ja menetelmiä sekä vaihtoehtoisia työkaluja ja niiden mahdollisia hyötyjä valittuihin verrattuina. Työkalujen valinta perustui pääasiassa yrityksen käytäntöön sekä omiin tottumuksiin ja osaamiseen.

## 2.1 Käytetyt työkalut

Tämä luku käsittelee sovelluksen kehitykseen käytettyjä ohjelmointimenetelmiä ja -työkaluja, kuten kieliä, kehyksiä ja kirjastoja. Verkkosivupohjaisena sovelluksena sivun rakenteen luontiin käytettiin HTML-merkintäkieltä ja ulkonäön määrittelyyn CSS-tyyliohjeistoa. JavaScriptiä käytettiin luomaan sivulle dynaamista sisältöä ja jQueryä muun muassa HTML-elementtien animointiin. Bootstrapin avulla luotiin modaalista sisältöä ja sivustosta saatiin responsiivinen useimmilla laitteilla. Tilastonäkymien kaaviot luotiin Google Charts -palvelun avulla modaalisten ikkunoiden päälle. Palvelinpuolen ohjelmistokielenä toimi PHP ja tietokantaohjelmistona MySQL. Ohjelmointiympäristönä käytettiin NetBeansia.

### 2.1.1 HTML

HTML eli Hypertext Markup Language on World Wide Web Consortiumin eli W3C:n ylläpitämä kuvauskieli, jolla kuvataan verkkosivujen rakennetta. Se on verkkosivujen olennaisin rakennetyökalu, jolla määritellään verkkosivujen sisällön elementtien, kuten otsikoiden, leipätekstin ja taulukoiden, rakennetta.

Hypertext eli hyperteksti tarkoittaa hyperlinkkejä sisältävää tekstiä. Hypertekstit ovat linkkejä, jotka voivat viitata esimerkiksi oman sivuston toiselle sivulle tai täysin ulkopuoliselle sivustolle.

Markup, eli merkintä, viittaa HTML:n tapaan merkitä verkkosivun rakennetta HTML-elementeillä, joita edustavat tagit. Tägeja ovat esimerkiksi `<title>`, `<h1>` ja `<p>`.

HTML:n kirjoittamista varten tarvitaan HTML-editori, joka voi olla kaikessa yksinkertaisuudessaan vain Windows-käyttöjärjestelmän oma Notepad-tekstieditori. Yksinkertainen esimerkki HTML-kielestä:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0"/>
    <title>Sivun otsikko</title>
  </head>
  <body>
    <h1>Kappaleen otsikko</h1>
    <p>Leipätekstiä</p>
    <a href="https://www.google.com">Hyperlinkki
Googlen-etusivulle</a>
  </body>
</html>
```



Kuva 1 näyttää miltä kyseinen esimerkki näyttäisi verkkoselaimessa.

## Kappaleen otsikko

Leipätekstiä

[Hyperlinkki Googlen-etusivulle](#)

Kuva 1. Yksinkertainen HTML-sivu

HTML-kuvauskieltä käytettiin eNollan sisällön rakenteen kuvaamiseen, koska se on Internet-selaimella käytettävä sovellus. HTML ei kuitenkaan yksinään riitä dynaamisen tai visuaalisesti näyttävän verkkosivun luontiin. HTML riittää kyllä yksinkertaisen, staattisen verkkosivun luontiin, mutta näyttävän ja dynaamisen sisällön luomiseen tarvitaan muita teknologioita, kuten CSS ja JavaScript.

### 2.1.2 CSS

CSS (Cascading Style Sheets) on W3C:n ylläpitämä, erityisesti verkkosivuja varten kehitetty tyyliohjeiden laji. CSS:ää käytetään yleensä HTML:n kanssa määrittelemään HTML-elementtien tyyliä kuten väri, koko, fontti ja asemointi. CSS tyyliohjeen voi kirjoittaa millä vain editorilla, kunhan tiedoston pääte vain on .css. Yksinkertainen esimerkki CSS-tyyliohjeesta kappaleen fontin, värin ja fonttikoon määrittelemiseksi:

```
p {
  font-family: verdana;
  font-size: 24px;
  color: green;
}
```

Kuva 2 näyttää miltä kappaleen 2.1.1 HTML-esimerkki (Kuva 1) näyttäisi verkkoselaimessa tämän tyyliohjeen kanssa.

## Kappaleen otsikko

Leipätekstiä

[Hyperlinkki Googlen-etusivulle](#)

Kuva 2. Yksinkertainen HTML + CSS -sivu

eNollan kehityksessä käytettiin CSS-tyyliohjeita sivuston visuaalisen ilmeen luomiseen. Jokaista asiakasta varten luodaan myös aina oma CSS-tiedosto, jossa on määritelty asiakkaan omat tyylit, kuten värit, yleensä samaan tapaan kuin asiakkaan omilla verkkosivuilla. Yhtiön yleiseen tapaan sivusto tunnistaa asiakkaan ID:n hänelle toimitetun URL:in sisältämän avaimen perusteella ja valitsee sen perusteella oikean CSS-tiedoston istunnolle.

Vakio CSS-tiedosto haetaan aina ennen vaihtoehtoista asiakastyyliä, joten jos asiakas ID:tä ei löydy niin käytetään oletustyyliä. Toiminnallisuus on toteutettu PHP:lla ja se suoritetaan HTML:n *<head>*-osan sisällä seuraavasti:

```
<?php
if(is_file("css/client_".($_SESSION["client_data"]["id"]).".css"))
{
    print '<link rel="stylesheet"
href="'.SITE_MAIN_URL.$base_url2.'css/client_'.($_SESSION["client_data"]["id"]).".css" />';
}
?>
```

### 2.1.3 JavaScript

JavaScript on Netscape Communications Corporationin kehittämä pääasiassa web-kehityksessä käytetty komentosarjakieli. Sen tärkeimpänä ominaisuutena pidetään dynaamisen toiminnallisuuden lisäämisen mahdollistamista verkkosivuille, johon sitä käytettiin myös eNollan kehityksessä. Sitä voidaan käyttää kuitenkin myös esimerkiksi työpöytä- tai mobiilisovellusten kehittämisessä.

JavaScript mahdollistaa verkkosivujen välittömän vastaamisen toimintaan lataamatta sivua uudelleen. Tällaisia toimintoja ovat esimerkiksi linkin tai painikkeen painallus sekä lomakkeen täyttö ja tarkastus. JavaScript saa web-sovelluksen tuntumaan hieman enemmän työpöytäsovellukselta. (McFarland 2014.)

JavaScriptiä voidaan kirjoittaa suoraan HTML-koodin sekaan, jolloin se tulee sijoittaa *<script>* ja *</script>* merkkien väliin. Vaihtoehtoisesti se voidaan kirjoittaa erilliselle .js päätteiselle tiedostolle, joka haetaan HTML-tiedostossa seuraavasti:

```
<script src="tiedostonimi.js"></script>
```

### 2.1.4 PHP

PHP (PHP: Hypertext Preprocessor) on palvelinpuolen web-ohjelmointikieli, joka sopii dynaamisten verkkosivujen luontiin. PHP on melko yksinkertainen kieli, joka tekee siitä sopivan myös aloittelijalle. Se kuitenkin tarjoaa paljon kehittyneitä ominaisuuksia myös kokeneemmille käyttäjille. PHP-koodia voidaan upottaa suoraan HTML-koodin sekaan sulkemalla se PHP:n aloitus- ja lopetusmerkkien, *<?php* ja *?>*, sisään. (The PHP Group 2017.)

PHP:n käyttö vaatii web-palvelimen asennettuna omalle tietokoneelle tai esimerkiksi web-hotellipalvelun, jolla on tuki PHP:lle. Osoitteesta <http://php.net/manual/en/install.php> löytyvät tarkemmat asennusohjeet.

Voidaan käyttää myös ohjelmia kuten XAMPP tai WampServer, jotka sisältävät yhdessä paketissa esimerkiksi Apache-palvelinohjelman, PHP-ohjelmointikielen ja MySQL-tietokantaohjelmiston. Yksinkertainen esimerkki PHP-upotteesta HTML-koodin seassa:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0"/>
    <title>Sivun otsikko</title>
  </head>
  <body>
    <h1>Kappaleen otsikko</h1>
    <p>Leipätekstiä</p>

    <?php
        echo "PHP-upote HTML-koodin seassa.";
    ?>

  </body>
</html>
```

### 2.1.5 jQuery

jQuery on erittäin suosittu JavaScript-kirjasto, joka sopii esimerkiksi verkkosivun erilaisten toimintojen käsittelyyn sekä HTML-elementtien valitsemiseen ja niiden animointiin. Sen tarkoitus on nopeuttaa ja yksinkertaistaa JavaScript-ohjelmointia. Kirjaston voi ladata osoitteesta <https://jquery.com/> ja sen saa käyttöönsä erillisenä paikallisena JavaScript-tiedostona viittaamalla siihen HTML:n `<script>`-tagilla seuraavasti:

```
<script src="jquery-3.2.1.min.js"></script>
```

Vaihtoehtoisesti se voidaan ottaa käyttöön CDN-palveluna viittaamalla jonkun julkisen palvelimen, kuten Googlen, jQuery-tiedostoon seuraavasti:

```
<script src="https://ajax.goog
leapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

### 2.1.6 Bootstrap

Bootstrap on erittäin suosittu responsiivisten verkkosivujen nopeampaa ja helpompaa luontia varten tarkoitettu ohjelmistokehys. Responsiivinen, eli mukautuva, verkkosivu tunnistaa millaisella ja minkä kokoisella laitteella sisältöä selataan ja skaalaa sisällön sen mukaiseksi ilman, että sivusta tarvitsee luoda erikseen erillinen mobiiliversio. Bootstrapia hyödynnettiin eNollassa juuri tähän tarkoitukseen, joten se on toimiva tuote erikokoisilla laitteilla matkapuhelimesta pöytätietokoneeseen.

Myös eNollan modaaliset, muun sisällön päälle hyppäävät näkymät, kuten tilastonäkymä, toteutettiin Bootstrapin liitännäisellä. Itse tilastotaulukot, jotka ovat modaalisessa näkymässä, toteutettiin Google Charts -palvelun avulla.

Bootstrapin voi ladata erillisenä itselleen tai vaihtoehtoisesti viitata siihen CDN-palvelun kautta. Osoitteesta <https://getbootstrap.com/> löytyy tarkemmat ohjeet asennukseen.

### 2.1.7 Google Charts

Google Charts on Googlen kehittämä JavaScript-kirjasto, jonka avulla voidaan luoda verkkosivuille näyttäviä, interaktiivisia kaavioita. Kirjasto tarjoaa monipuolisesti erilaisia pohjavaihtoehtoja kuten pylväs-, viiva-, piirakka- ja pistekaavioita. Kaavioita voi muokata pitkälle oman tarpeensa mukaan. Alla oleva koodiesimerkki (Kuva 3) näyttää kuinka saadaan aikaan yksinkertainen piirakkakuvi (Kuva 4) HTML-tiedostossa JavaScriptin avulla.

```
<html>
<head>
  <!--Load the AJAX API-->
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">

    // Load the Visualization API and the corechart package.
    google.charts.load('current', {'packages':['corechart']});

    // Set a callback to run when the Google Visualization API is loaded.
    google.charts.setOnLoadCallback(drawChart);

    // Callback that creates and populates a data table,
    // instantiates the pie chart, passes in the data and
    // draws it.
    function drawChart() {

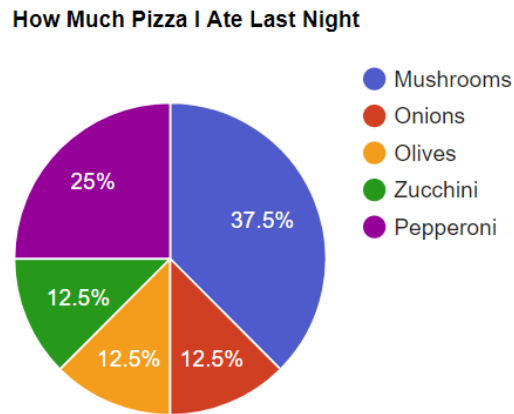
      // Create the data table.
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Topping');
      data.addColumn('number', 'Slices');
      data.addRows([
        ['Mushrooms', 3],
        ['Onions', 1],
        ['Olives', 1],
        ['Zucchini', 1],
        ['Pepperoni', 2]
      ]);

      // Set chart options
      var options = {'title':'How Much Pizza I Ate Last Night',
                    'width':400,
                    'height':300};

      // Instantiate and draw our chart, passing in some options.
      var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>

<body>
  <!--Div that will hold the pie chart-->
  <div id="chart_div"></div>
</body>
</html>
```

Kuva 3. Piirakkakuvi koodiesimerkki (Google Inc. 2017).



Kuva 4. Piirakkakuvi (Google Inc. 2017).

eNollan tilastonäkymissä käytettiin Google Charts -palvelulla tuotettuja viiva- ja pylväskaavioita sekä niiden yhdistelmää. Tilastonäkymä käydään läpi luvussa, jossa käsitellään tuotteen ominaisuudet ja näkymät.

### 2.1.8 MySQL

MySQL on suosittu avoimen lähdekoodin tietokantaohjelmisto, jota ylläpitää Oracle Corporations. MySQL on suosittu etenkin web-palvelujen kannana. Esimerkiksi Facebook on yksi kannan merkittävimmistä käyttäjistä. eNollan kehityksessä käytettyä PHP-ohjelmointikieltä käytetään usein yhdessä MySQL-tietokannan kanssa.

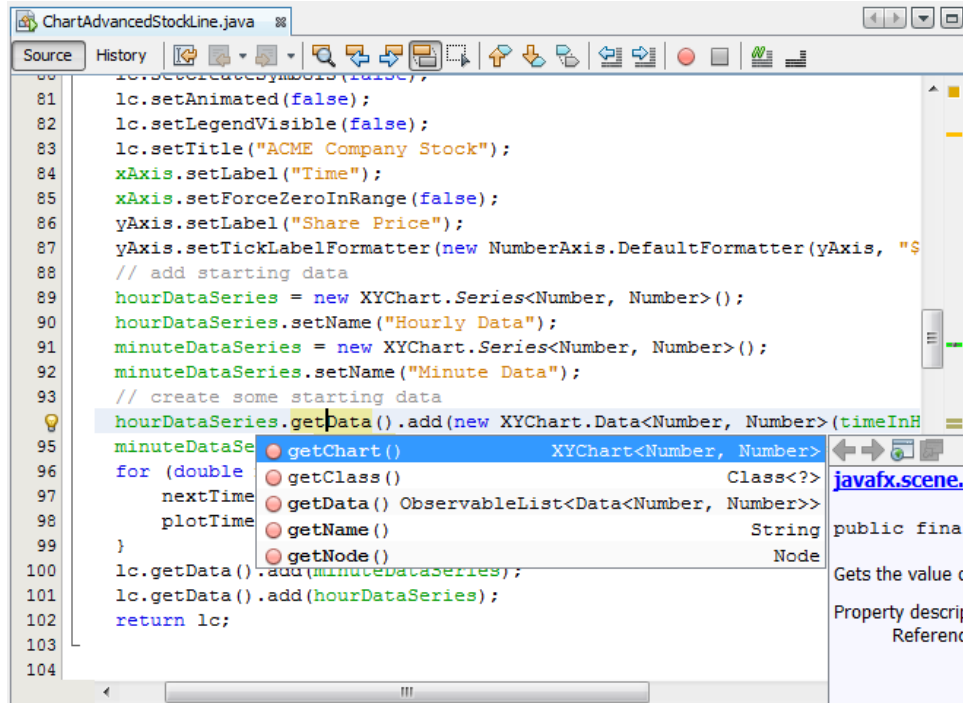
Tietokanta on rakenne, joka koostuu siihen tallennetusta varsinaisesta tiedosta ja metatiedosta. Jos tietokantaan tallennetaan esimerkiksi ihmisten yhteystietoja niin varsinainen tieto voi olla esimerkiksi nimiä, osoitteita ja puhelinnumeroita. Metatieto on niin sanotusti tietoa tiedosta. Se kuvailee tiedon ominaisuuksia kuten jokaisen tietoelementin tyyppin. Tyyppejä voi olla esimerkiksi teksti, numeerinen tai päivämäärä. Tietokantaohjelmistolla hallitaan tietokantojen rakennetta ja pääsyä tietoon. (Coronel & Morris 2016, 6.)

MySQL-tietokantaohjelmisto voidaan asentaa erillisenä tietokoneelle tai esimerkiksi XAMPP-ohjelmiston mukana vastaavasti kuin PHP. Osoitteesta <https://dev.mysql.com/doc/refman/5.7/en/installing.html> löytyy tarkemmat ohjeet asennukseen. MySQL-tietokantaa käytetään eNollan yhteydessä mittarien parametrien ja mittareihin syötettyjen tietojen säilyttämiseen.

### 2.1.9 NetBeans

NetBeans IDE on ilmainen, avoimen lähdekoodin integroitu ohjelmointiympäristö. Siinä on hyvät työkalut web-sovelluskehitykseen, joten se sopii hyvin eNollan ohjelmointiympäristöksi. NetBeans on kirjoitettu Javalla,

mutta se tukee myös esimerkiksi C, C++, PHP, HTML, JavaScript ja CSS -kieliä. Sen tekstieditori (Kuva 5) sisältää työkaluja koodin kirjoittamisen ja lukemisen nopeuttamiseksi ja helpottamiseksi. Niitä ovat muun muassa koodin refaktorointi ja täydennys, pop-up -ohjeet sekä syntaktinen ja semanttinen koodin korostus. (Oracle Corporation n.d.)



Kuva 5. Fast & Smart Code Editing (Oracle Corporation n.d.).

NetBeansin asentamiseksi vaaditaan Java SE Development Kit 8, jonka voi ladata osoitteesta <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. PHP-kehittämistä varten tarvitaan myös Java Runtime Environment, joka sisältyy PHP:tä tukeviin NetBeans-versioihin. Itselleen sopivan NetBeans-paketin voi ladata osoitteesta <https://netbeans.org/downloads/>.

## 2.2 Vaihtoehtoisia työkaluja

Tämä luku käsittelee joitain suosituimpia vaihtoehtoisia työkaluja, joita eNollan kehitykseen olisi voinut käyttää käytettyjen menetelmien tilalla tai tukena. Pohditaan myös niiden hyviä ja huonoja puolia käytettyihin menetelmiin verrattuina.

### 2.2.1 Python

Python on avoimen lähdekoodin moniparadigmainen ohjelmointikieli. Se tukee olio-, funktionaalisia- ja proseduraalisia ohjelmointiparadigmoja. Pythonin ominaisuuksiin kuuluu helppolukuinen koodi ja kirjastojen toi-

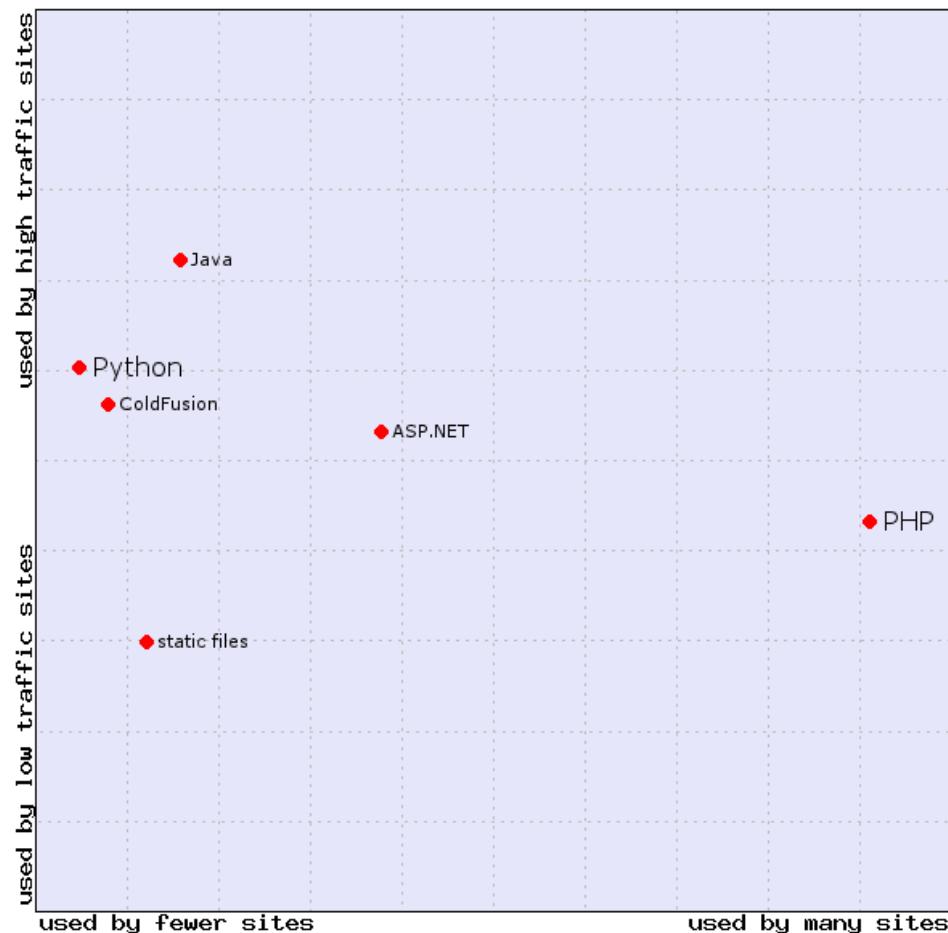
mintä sekä suunnittelu, joka tehostaa tuotteliaisuutta, laatua, mukautuvuutta eri ympäristöihin sekä komponenttien integraatiota. Sitä pidetään yhtenä maailman käytetyimmistä ohjelmointikielistä. (Lutz 2014, 1.)

Pythonia pidetään joustavampana, monikäyttöisempänä ja natiivisti turvallisempänä kuin PHP:ta. Se tukee myös työpöytäsovellusten kehitystä ja lisäksi kaikki suurimmat mobiilialustat tukevat Pythonilla kehittämistä. PHP:n etuna voidaan kuitenkin pitää sitä, että se suunniteltiin juuri web-kehitystä varten. (Hornostaiev 2017.)

Jos vertaillaan PHP:tä ja Pythonia niitä käyttävien verkkosivujen määrän mukaan niin PHP on kuitenkin huomattavasti Pythonia suosituampi palvelinpuolen ohjelmointikieli. Vuonna 2017 PHP:tä käytti yli 80 % ja Pythonia noin 0.2 % verkkosivuista. Vertailussa tulee kuitenkin ottaa huomioon, että Python on suosituampi kieli korkealiikenteisillä sivustoilla. (W3Techs 2017.)

Suurimpia Pythonin käyttäjiä ovat muun muassa Google, YouTube, Instagram, Pinterest ja NASA (World Web Technology 2017). Alla olevasta kaaviosta (Kuva 6) näkee Pythonin ja PHP:n suhteen niitä käyttävien verkkosivujen määrän ja niiden liikenteen mukaan.

Market position, selected server-side programming languages, 27 Nov 2017, W3Techs.com



Kuva 6. Market position, selected server-side programming languages, 27 Nov 2017, W3Techs.com (W3Techs 2017.)

PHP valikoitui eNollan web-ohjelmointikieleksi omasta tottumuksesta ja toimeksiantajan käytännöstä johtuen. Muita suosittuja palvelinpuolen ohjelmointikieliä ovat muun muassa ASP.NET, Java, Ruby ja Perl.

### 2.2.2 Laravel

Laravel on avoimen lähdekoodin MVC-arkkitehtuuriin perustuva ohjelmistokehys PHP:lle, jonka tarkoituksena on lisätä kehitysprosessin mukavuutta ja yksinkertaisuutta. Se on yksi suosituimmista PHP:n ohjelmistokehyksistä. Laravelin suurimpia vahvuuksia on muun muassa sen yksinkertainen reitityksen käsittely, turvallisuus, tietokantatyökalut, käyttäjän todentaminen, session käsittely, tiedostonhallinta ja poikkeuksen käsittely. (Traversy Media 2017.)

Ohjelmistokehykset voivat pitkällä aikavälillä helpottaa ohjelmointityötä. PHP:n ohjelmistokehykset eivät olleet itselleni tuttuja aloittaessani eNollan kehityksen, joten tästä sekä yhtiön käytännöstä johtuen niiden käyttöön ei PHP:n kohdalla päädytty. Laravelin lisäksi muita suosittuja PHP:n ohjelmistokehyksiä ovat muun muassa Symfony, CodeIgniter ja CakePHP.

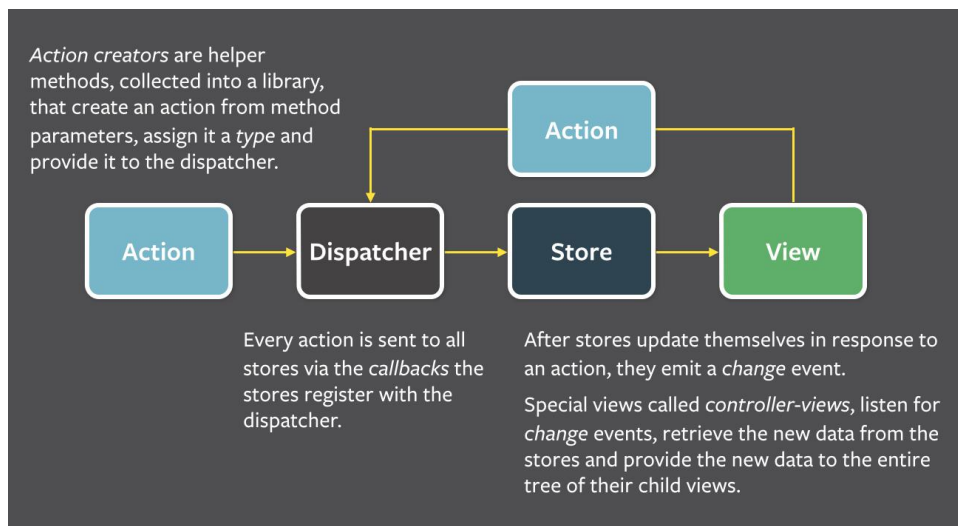
### 2.2.3 ReactJS

ReactJS on käyttöliittymien rakennusta varten tarkoitettu JavaScript-kirjasto, jota ylläpitää muun muassa Facebook ja Instagram. Se mahdollistaa web-sovellusten luonnin, joiden sisältö voi muuttua lataamatta sivua uudelleen.

ReactJS kuvaa itseään MVC-arkkitehtuurin V- eli View-osaksi. Se on JavaScript-kirjasto, joka on yhteensopiva monien JavaScript-ohjelmistokehysten tai jopa muiden JavaScript-kirjastojen kanssa. Reactia käytetään usein omillaan yhdistettynä Flux-datavirtamallin kanssa. (Robbestad 2016, 2.)

Flux on muun muassa Facebookin asiakaspuolen web-sovelluksiin käytämä ohjelmistoarkkitehtuurimalli. Se täydentää Reactin view-komponentteja käyttämällä hyödyksi yksisuuntaista datavirtausta, jota Kuva 7 esittää. (Facebook Inc. n.d.)





Kuva 7. Flux Structure and Data Flow (Facebook Inc. 2016).

ReactJS on hyvä ratkaisu, kun rakennetaan suuria sovelluksia, jotka sisältävät jatkuvasti muuttuvaa dataa. eNollan kehitykseen riitti kuitenkin hyvin jo aiemmin itselleni osittain tuttu jQuery, joka teki HTML-elementtien hallinnasta ja JavaScriptin kirjoittamisesta sujuvampaa. Muita vaihtoehtoja Reactille on muun muassa Angular, Vue ja Preact.

#### 2.2.4 MariaDB

MariaDB on avoimen lähdekoodin relaatiotietokantajärjestelmä, joka haavautui MySQL:stä sen luojaan, Michael Wideniuksen, toimesta. MariaDB sisältää samat perusominaisuudet kuin muutkin relaatiotietokantajärjestelmät. Se voi muokata dataa ja sen rakennetta sekä suorittaa sille kyselyjä. Kaiken tyyppiset ohjelmistot voivat keskustella MariaDB:n kanssa hallitakseen dataa. Lausekkeet kirjoitetaan SQL-kielellä. (Emilien & Razzoli 2015, 1.)

MariaDB on alaspäin yhteensopiva MySQL:n kanssa, joka tarkoittaa useimmissa tapauksissa sitä, että MariaDB on toimiva asennettuna suoraan MySQL:n tilalle (MariaDB 2017).

MariaDB:n on luvattu pysyvän avoimen lähdekoodin kantana, jonka vuoksi monet pitävät sitä luotettavampana kuin kaupallisista tuotteistaan tunnetun Oraclen omistuksessa oleva MySQL. Sen tärkeimpiä käyttäjiä ovat muun muassa Google, Wikipedia ja Booking.com. (Hovi 2017.)

MySQL:ään pohjautuvana MariaDB on hyvin samankaltainen kuin MySQL, sisältäen kuitenkin joitain pieniä parannuksia muun muassa kyselykieleen ja sisäiseen arkkitehtuuriin. MySQL valikoitui eNollan tietokantajärjestelmäksi yhtiön käytännöstä johtuen. Se oli myös itselleni tutumpi kuin MariaDB.

### 2.2.5 Eclipse

Eclipse on ilmainen avoimen lähdekoodin ohjelmointiympäristö, joka tukee ensisijaisesti Javaa, mutta liitännäisten avulla myös muun muassa C, C++, C#, JavaScript, Perl, PHP, Python ja Ruby -ohjelmointikieliä. Enemmän standardoituun Oracle Corporationin omistamaan NetBeansiin verrattuna Eclipse on monine liitännäisineen enemmän käyttäjän mukautettavissa omiin tarpeisiinsa.

Eclipse on perspektiivipohjainen. Perspektiivi Eclipsessä on kokoelma näkymiä, editoreita ja työkalupalkkeja, joiden asettelua voidaan vaihtaa tietyn tehtävän edun mukaiseksi. Asettelu tallentuu kyseisen perspektiivin alle sitä muokattaessa, jolloin sitä voi hyödyntää uudestaan, kun suoritetaan vastaavaa tehtävää. Eclipsessä on valmiiksi oletusperspektiivejä esimerkiksi Java-ohjelmien kehitykseen sekä virheiden diagnosointiin ja korjaamiseen. (Burnette 2005, 13.)

Sekä Netbeans että Eclipse olivat eNollan kehityksen kannalta hyvin samantapaiset ohjelmointiympäristöt. NetBeans valikoitui ohjelmistoympäristöksi yhtiön käytännöstä johtuen sekä siksi, että se oli itselleni jo entuudestaan tuttu.

## 3 TUOTTEEN SUUNNITTELU

Tuote suunniteltiin ensin Flowmedik Oy:n sisäisesti sairaalarakentamisen näkökulmasta. Yhtiöllä on omistajina, työntekijöinä sekä muina yhteistyökumppaneina useita sosiaalityön ja terveydenhuollon ammattilaisia, jotka tunsivat alan tarpeet ja joiden kanssa tuotteen ensimmäisen version tarpeet määriteltiin.

Mitään varsinaista projektisuunnitelmaa ei tuotteen suunnittelusta kuitenkaan tehty. Tuotteesta kehitettiin ensin yhtiön asiantuntijoiden näkemyksien mukainen versio, jonka jälkeen tuotetta määriteltiin vielä lisää asiakkaiden edustajien kanssa heidän tarpeidensa mukaan. Omaan työhöni asiakkaan kanssa suunnitteluvaiheessa kuului heidän haluamien ominaisuuksien ja visuaalisten toiveiden huomioon ottaminen ja kehittäminen. Lisäksi minun tuli määrittää heidän kanssaan tiedot, joita itse tarvitsen heidän organisaatioiltaan, jotta voin luoda heille heidän toiveidensa mukaisesti toimivat toiminnanohjausmittarit sovellukseen.

Suunnittelun ja kehityksen jälkeen sekä jo sen aikana tuote oli testikäytössä muutamalla asiakkaalla. Tuotetta jatkokehitettiin asiakkaille syntyneiden tarpeiden mukaan sitä mukaa kun niitä tuli. Ne olivat pääosin visuaalisuuteen ja tilastonäkymiin liittyviä pieniä muutoksia. Toimeksiantajan

toimitavan mukaan tuotetta tullaan määrittelemään lisää vielä potentiaalisten uusien asiakkaiden tarpeiden mukaan, jolloin tuotteen kehitys tulee vielä todennäköisesti jatkumaan tulevaisuudessa.

## 4 OMINAISUUDET JA NÄKYMÄT

Tämä luku käsittelee eNollan ominaisuuksia ja näkymiä sekä niiden käyttötarkoitusta ja niiden tarjoamaa informaatioarvoa. Kuvissa käytetään esimerkkinä kuvitteellista esittelytarkoitukseen luotua ”Polikäynnit”-tapausta, johon asetetut parametrit avataan tässä luvussa.

### 4.1 Case: Polikäynnit

Sairaalan poliklinikoita uudisrakennetaan, jotta voitaisiin vastaanottaa enemmän poliklinikkakäyntejä tulevaisuudessa. Otetaan tavoitteeksi lisätä poliklinikkakäyntejä 20 % vuoden 2018 loppuun mennessä. Kehitystyö on aloitettu jo vuoden 2015 alusta. Jotta rahoittajalle luvattuun tavoitteeseen voitaisiin päästä määräajassa, tuloksia pitäisi olla jo syntynyt. Otetaan käyttöön eNolla-mittari kehitysvaiheen tavoitteiden seuraamista varten.

### 4.2 Päänäkymä

Päänäkymä (Kuva 8) on eNolla-mittarin etusivulla (LIITE 1), jonka asiakas näkee siirtyessään tuotteesta luodulle asiakaskohtaiselle verkkosivulleen. Päänäkymästä voidaan luoda myös upote asennettavaksi esimerkiksi asiakkaan omille verkkosivuille tai näyttöpäätteelle fyysisiin tiloihin. Asiakkaalle voidaan luoda useita eri mittareita.

Päänäkymä sisältää

- asiakkaan logon
- mittarikohtaisen infokentän
- mittarivalikon
- pääluvun, joka kertoo tavoitteiden toteutumisesta
- tilastopainikkeen
- lisäasetukset -painikkeen käyttäjille, joilla on oikeus hallita mittareita.



Kuva 8. Päänäkymä

#### 4.3 Infokenttä

Valitun mittarin infokenttä (Kuva 9) kertoo mittariin asetetut parametrit, joiden mukaan muun muassa lasketaan tavoitteet.

Infokenttä sisältää

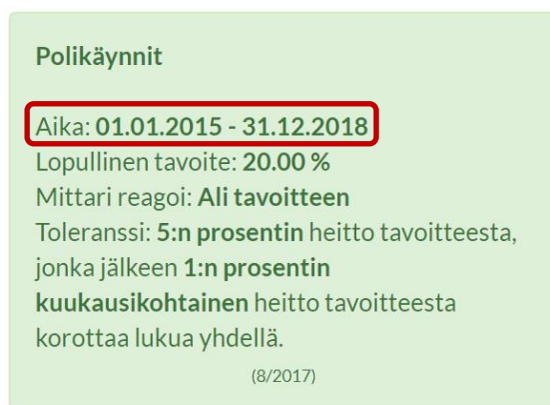
- seurattavan mittarin nimen
- projektille asetetun kehitysvaiheen ajan
- seurattavalle mittarille asetetun tavoitteen
- mittarin reagointisuunnan
- mittarille asetetun toleranssin
- luvun, joka kasvattaa mittarin päänäkymän numeroa yhdellä
- tiedon, millä aikavälillä mittaria seurataan
- viimeisimmän päivitysajankohdan.



Kuva 9. Infokenttä

#### 4.3.1 Kehitysaika

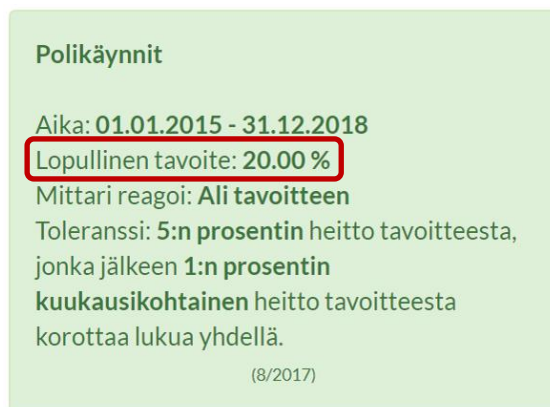
Infokentän aikaparametri (Kuva 10) kertoo seurattavalle projektille asetetun kehitysaajan. Alkupäivämäärä kertoo mistä seurattavan kohteen kehitys on lähtenyt käyntiin. Loppupäivämäärä kertoo milloin kehitysvaiheen olisi pitänyt saavuttaa tulevan tasaisen vaiheen toiminnan tavoitteellinen teho-taso. Tasainen vaihe tarkoittaa aikaa kehitysaajan jälkeen, jolloin toiminta jatkuu tasaisena tavoitteeseen päästyä.



Kuva 10. Kehitysaika

#### 4.3.2 Tavoite

Infokentän tavoiteparametri (Kuva 11) kertoo seurattavalle projektille asetetun tehostustavoitteen. Tavoite tarkoittaa parannusta, joka pitäisi saavuttaa mitattavan yksikön alkuarvoihin, kuten euroihin tai suoritteisiin, kehitysvaiheen loppuun mennessä. Tavoite voidaan asettaa prosentuaalisena tai yksikköparannuksena ja se nousee lineaarisesti kehitysvaiheen alusta loppuun.

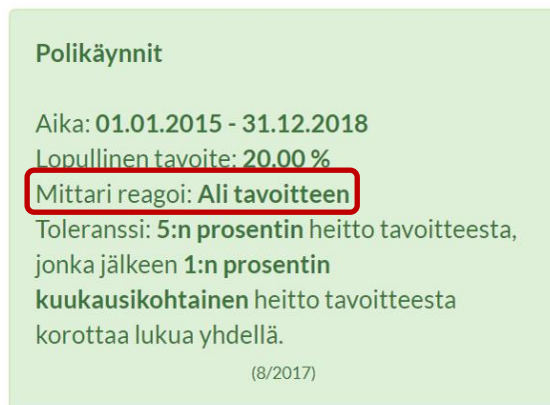


Kuva 11. Tavoite

Ohjelma luo asetettujen parametrien perusteella taustalla taulukon, joka sisältää tasaisesti nousevat tavoitteet aikavälille. Viimeinen, eli kehitysvaiheen lopun tavoite taulukossa on mittariin asetettu lopullinen tavoite.

#### 4.3.3 Reagointisuunta

Infokentän reagointisuuntaparametri (Kuva 12) kertoo seurattavan projektin kehityksen positiivisen suunnan. Mittari reagoi asetuksista riippuen joko yli tai ali tavoitteen. Reagointisuunta kertoo, onko tavoitteelle asetettuja lähtöarvoja tarkoitus pienentää vai suurentaa.



Kuva 12. Reagointisuunta

#### 4.3.4 Toleranssi

Mittareihin voidaan asettaa raja-arvo, jonka sisällä päänäkömäärän luku ei vielä reagoi syötettyihin toteumiin, vaikka ei oltaisiakaan varsinaisessa tavoitteessa. Infokentän toleranssiparametri (Kuva 13) kertoo seurattavan projektin tavoitteelle asetetun toleranssiarvon, joka voidaan asettaa joko prosentteina tai yksiköinä.

**Polikäynnit**

Aika: 01.01.2015 - 31.12.2018  
 Lopullinen tavoite: 20.00 %  
 Mittari reagoi: **Ali tavoitteen**  
**Toleranssi: 5:n prosentin** heitto tavoitteesta,  
 jonka jälkeen **1:n prosentin**  
**kuukausikohtainen** heitto tavoitteesta  
 korottaa lukua yhdellä.

(8/2017)

Kuva 13. Toleranssi

#### 4.3.5 Poikkeamaluku

Infokentän poikkeamalukuparametri (Kuva 14) kertoo kuinka suuri heitto tavoitteesta nostaa päänäkymän punaista lukua (Kuva 15) yhdellä. Poikkeamaluku voidaan asettaa joko prosentteina tai yksiköinä.

**Polikäynnit**

Aika: 01.01.2015 - 31.12.2018  
 Lopullinen tavoite: 20.00 %  
 Mittari reagoi: **Ali tavoitteen**  
 Toleranssi: 5:n prosentin heitto tavoitteesta,  
 jonka jälkeen **1:n prosentin**  
**kuukausikohtainen** heitto tavoitteesta  
 korottaa lukua yhdellä.

(8/2017)

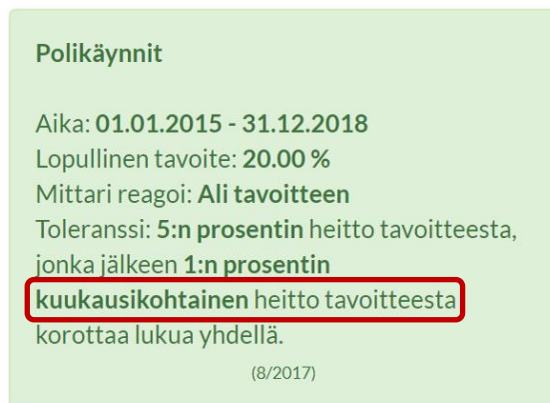
Kuva 14. Poikkeamaluku



Kuva 15. Päänäkymän luku

#### 4.3.6 Seuranta-aikaväli

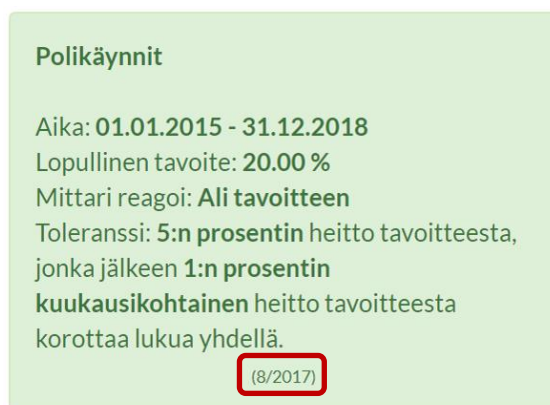
Infokentän seuranta-aikaväliparametri (Kuva 16) kertoo, seurataanko mittariin syötettäviä lukuja päivä-, kuukausi-, vai esimerkiksi vuositasolla. Tehostettavat lähtöarvot asetetaan valitun aikavälin mukaan. Kuukausitason seurannassa asetetaan siis jokaiselle kuukaudelle oma lähtöarvo, jota tulisi parantaa tavoitteen verran kehitysvaiheen loppuun mennessä, joka on esimerkiksi 20 prosenttia. Esimerkiksi poliklinikkatoiminnassa eri kuukausien välillä voi olla suuria eroja toiminnan määrässä, jolloin tämän tapainen toiminnan tehostamisen seuranta on järkevää.



Kuva 16. Seuranta-aikaväli

#### 4.3.7 Viimeisin päivitys

Infokentän viimeinen rivi (Kuva 17) kertoo käyttäjälle, mikä on päivämäärältään mittarin uusin merkintä. Päivämäärän avulla käyttäjä voi tulkita kuinka ajankohtaista tietoa mittari sisältää tarkasteluhetkellä. Päivämäärä ei siis määräydy tiedon syöttöajankohdan, vaan syöttöhetkellä tiedolle määrätyn päivämäärän mukaan.

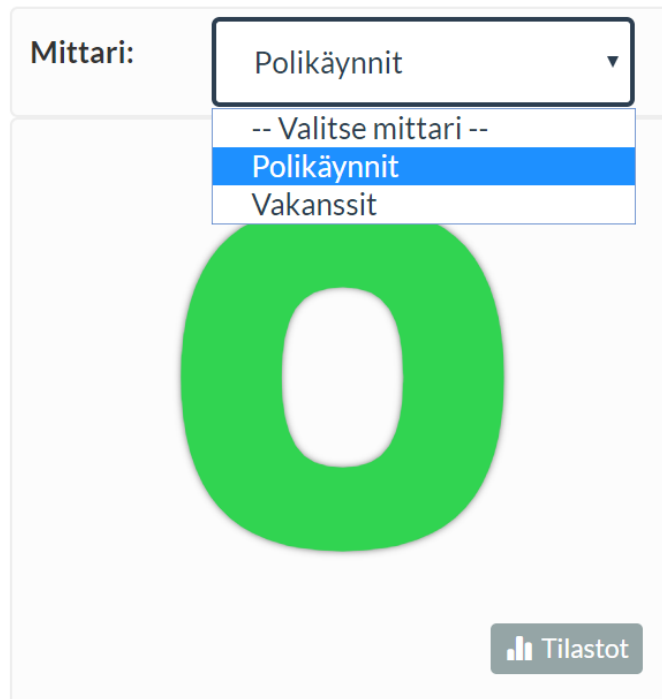


Kuva 17. Päivitysaika



#### 4.4 Mittarivalikko

Sovelluksen käyttäjälle voidaan luoda useita eri mittauskohteita. Tarkasteltavan mittarin voi valita pääluvun yläpuolella olevasta pudotusvalikosta (Kuva 18).



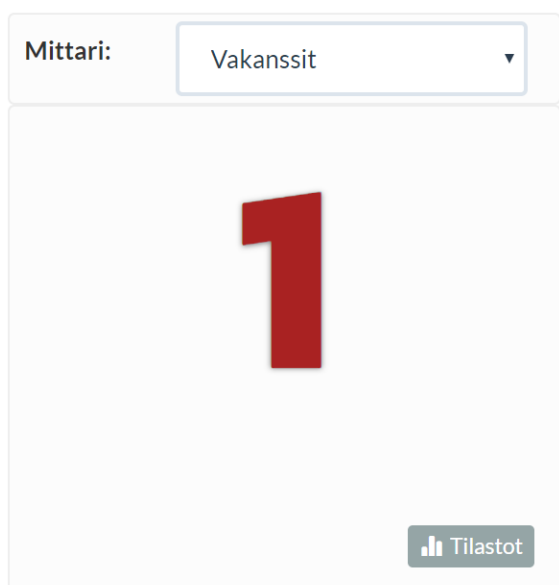
Kuva 18. Mittarivalikko

#### 4.5 Päänäkymän luku

Päänäkymän luku kertoo käyttäjälle yksinkertaisella visuaalisella tavalla, kuinka lähellä tavoitetta ollaan tarkasteluhetkellä. Vihreä nolla (Kuva 19) mittarissa tarkoittaa sitä, että viimeisimmän syötetyn merkinnän mukaan ollaan tavoitteessa ja punainen luku (Kuva 20) sitä, että viimeisimmän syötetyn merkinnän mukaan ei olla tavoitteessa. Mitä suurempi punainen luku on, sitä kauempana tavoitteesta ollaan. Luku kasvaa mittariin asetettujen parametrien, kuten reagointiluvun ja toleranssin, mukaan. Tarkasteluhetkikohtainen tavoite määräytyy viimeisimmän merkinnän päivämäärän mukaan. Luvun alla olevalla Tilastot-painikkeella pääsee tarkastelemaan mittariin syötettyjä merkintöjä tarkemmin.



Kuva 19. Vihreä nolla



Kuva 20. Punainen luku

#### 4.6 Tilastot

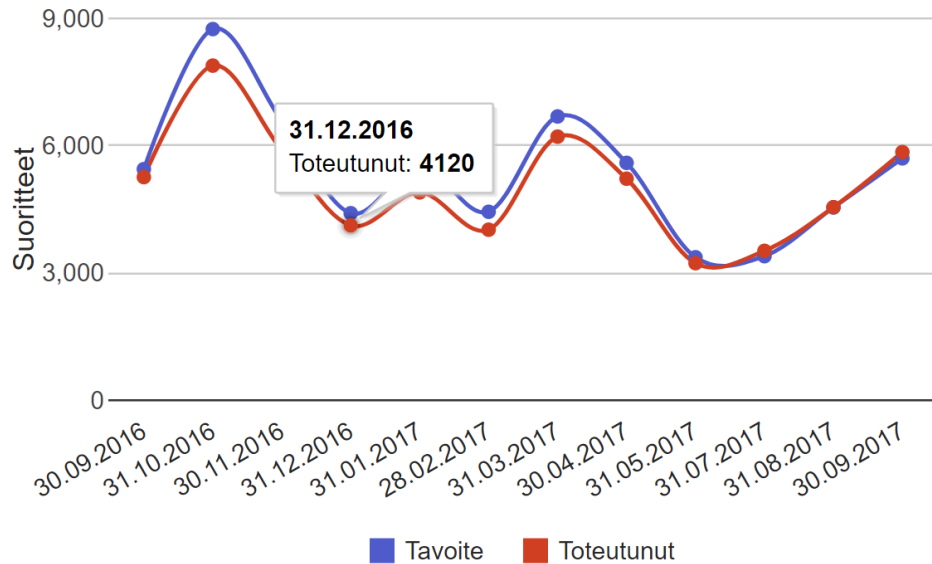
Mittarin päänäkymän luku kertoo tarkasteluhetken aikaisen tilanteen suhteessa sen aikaiseen tavoitteeseen. Kumulatiivinen historiatieto löytyy tilastonäkymästä. Näkymä toteutettiin Bootstrapin modaalinäkömaliitännäisellä, eli sivuston päälle hyppäävällä ikkunalla, jonka päälle luotiin kaavioita Google Charts -palvelun avulla (LIITE 2).

Tilastoissa on kolme erilaista kaaviota:

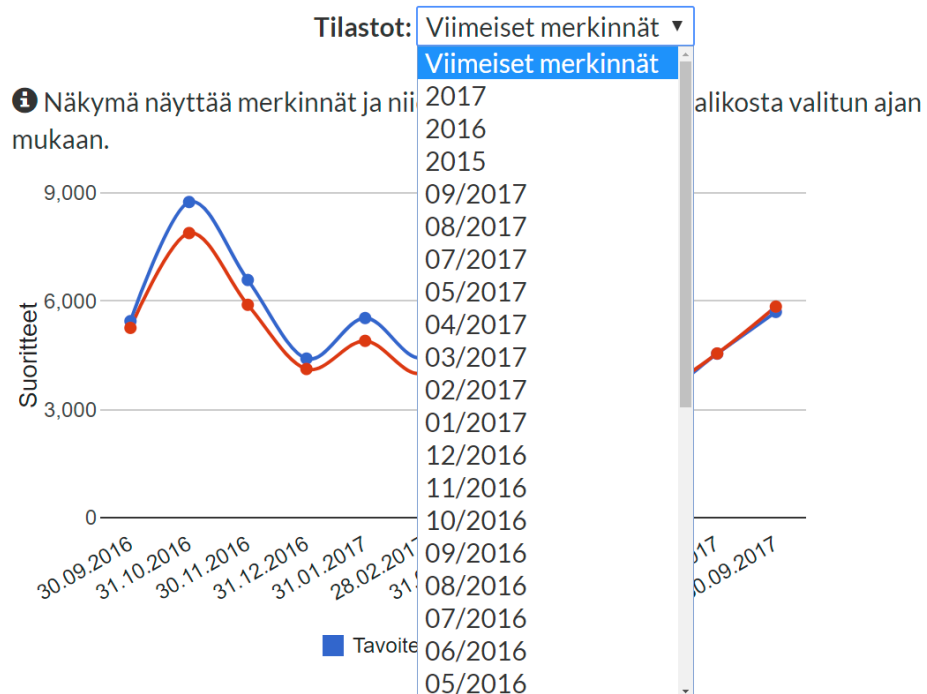
- kaikki merkinnät -kaavio
- tarkasteluajavälikohtainen keskiarvopylväskaavio
- projektin aikajana -kaavio.

#### 4.6.1 Kaikki merkinnät -kaavio

Tilastonäkymän ensimmäisestä kaaviosta (Kuva 21) näkee viimeisimmät merkinnät ja niiden tavoitteet. Kaavion yläpuolella olevasta pudotusvalikosta voi valita merkinnät mittarin tarkasteluajavälin mukaan päivä-, kuukausi- tai vuosikohtaisesti (Kuva 22).



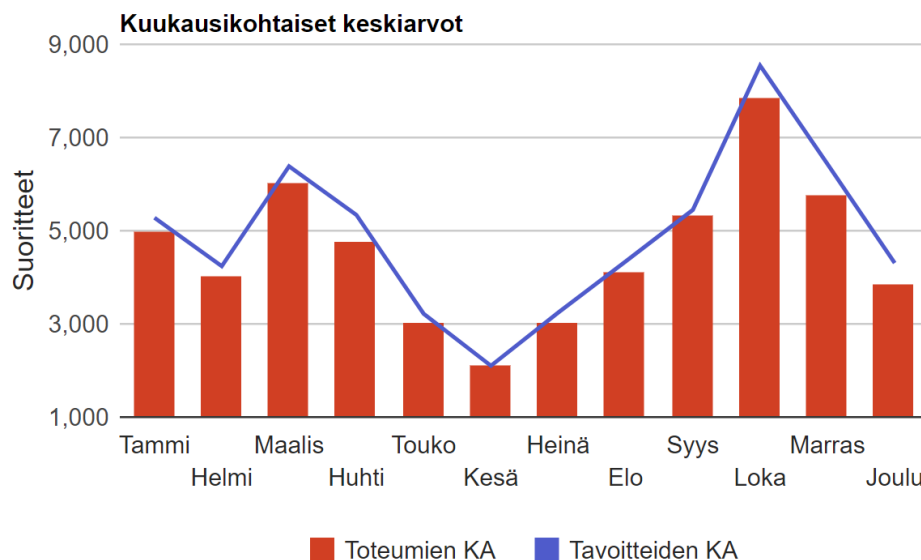
Kuva 21. Kaikki merkinnät -kaavio



Kuva 22. Tarkasteltava aikaväli

#### 4.6.2 Keskiarvokaavio

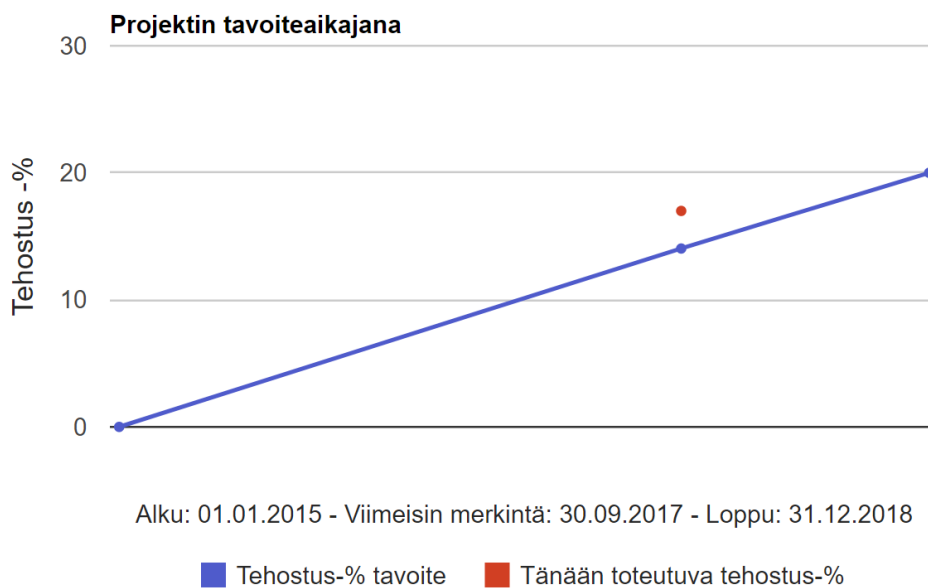
Tilastonäkymän toisesta kaaviosta (Kuva 23) näkee pylvänä kaikkien merkintöjen keskiarvot ja viivana niiden tavoitteiden keskiarvot. Keskiarvot lasketaan mittariin asetetun tarkasteluajavälin mukaan.



Kuva 23. Keskiarvokaavio

#### 4.6.3 Aikajanakaavio

Tilastonäkymän kolmannesta kaaviosta (Kuva 24) näkee asetetun kehitysvaiheen aikajanana. Kaavion sininen jana kuvaa lineaarisesti nousevaa mittariin asetettua tavoitetta kehitysvaiheen alusta loppuun. Sininen piste janaalla on päivämäärällisesti viimeisimmän merkinnän kohdalla. Vertikaalisesti samalla kohdalla oleva punainen piste kuvaa viimeisimmän merkinnän toteumaa ja sen suhdetta lineaarisesti nousevaan tavoitteeseen eli siniseen janaan.







Kuva 24. Aikajanakaavio

#### 4.7 Lisäasetukset

Käyttäjille, joilla on mittarinhallinta- ja merkintöjen lisäsoikeudet, näkyy päänäköymässä ylimääräinen painike, jonka alta löytyy lisäasetuksia (Kuva 25). Valikon kautta käyttäjä voi kopioida leikepöydälle suoran linkin valittuun mittariin ja sen upotenäkymään sekä avata yksinkertaisen listan kaikista merkinnöistä, jotka mittariin on syötetty. Listan kautta käyttäjä voi poistaa merkintöjä esimerkiksi virheellisen merkinnän sattuessa (Kuva 26).



Kuva 25. Lisäasetukset

Päivämäärä	Arvo	Poista
31.01.2015	5010.00	
28.02.2015	3800.00	
31.03.2015	5900.00	
30.04.2015	4080.00	
31.05.2015	2700.00	
30.06.2015	2400.00	
31.07.2015	3120.00	
31.08.2015	3650.00	

Kuva 26. Lista kaikista merkinnöistä

#### 4.8 Upotenäkymä

Kaikista mittareista voidaan luoda upotteet asennettaviksi esimerkiksi yksikön intraan tai tiloihin erilliselle näytölle. Upote on mittarista luotu yksinkertaistettu näkymä (Kuva 27). Upotetta klikkaamalla käyttäjä pääsee käsiin kyseisen mittarin tietoihin. Upotenäkymästä voidaan luoda erilaisia versioita asiakkaan toiveiden mukaisesti.



### Polikäynnit

Aika: 01.01.2015 - 31.12.2018

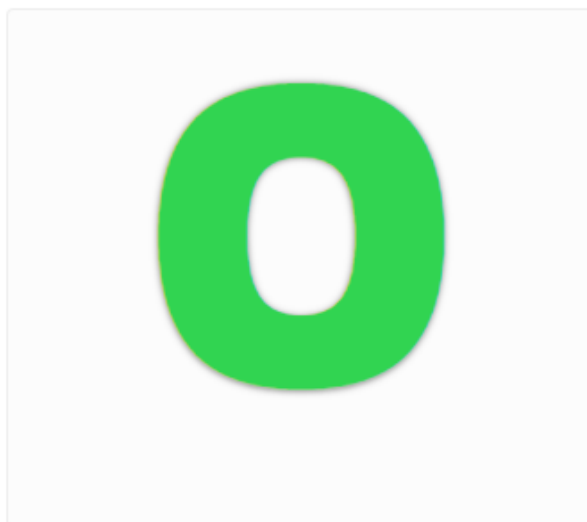
Lopullinen tavoite: 20.00 %

Mittari reagoi: **Ali tavoitteen**

Toleranssi: 5:n prosentin heitto tavoitteesta, jonka jälkeen 1:n prosentin

kuukausikohtainen heitto tavoitteesta korottaa lukua yhdellä.

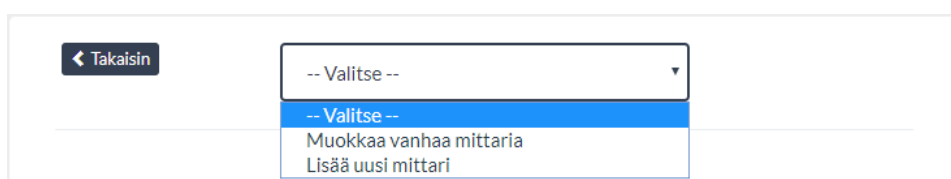
(8/2017)



Kuva 27. Upotenäkymä

#### 4.9 Mittarinhallintatyökalut

Mittarinhallintaoikeudet omaava käyttäjä pääsee käsiksi mittarinhallintatyökaluihin navigointipalkin kautta. Siirtyessä mittarinhallintaan tulee ensin valita pudotusvalikosta, halutaanko luoda uusi mittari vai muokata vanhaa (Kuva 28).



Kuva 28. Mittarihallintatoiminnon valinta

#### 4.9.1 Uuden mittarin luonti

Uuden mittarin luonnin valinta avaa näkymän, johon syötetään uuden mittarin parametrit (Kuva 29). Mittariin tulee määrittää

- mittarin nimi
- kehitysvaiheen alkupäivämäärä
- kehitysvaiheen loppupäivämäärä
- parannettavat lähtöarvot ja seuraamisen aikajako
- tehostustavoiteprosenttiyksikkö
- mittarin reagointisuunta
- sallittu toleranssi
- luku, joka nostaa mittarin lukemaa yhdellä
- heiton ja toleranssin tyyppi.

Parannettavat lähtöarvot ja seuraamisen aikajako valitaan sen mukaan, millä aikavälillä tarkasteltavaa kohdetta halutaan seurata. Mittaria voidaan päivittää esimerkiksi kuukausi- tai päiväkohtaisesti. Jos valitaan seurattavaksi aikajaoksi kuukausikohtainen, tulee parannettaviin lähtöarvoihin asettaa jokaiselle vuoden kuukaudelle, tammikuusta joulukuuhun, oma lähtöarvo, jonka mukaan tavoitteet lasketaan taustalla muiden asetettavien parametrien mukaan.



### Lisää mittari

Mittari reagoi jos kehitys:

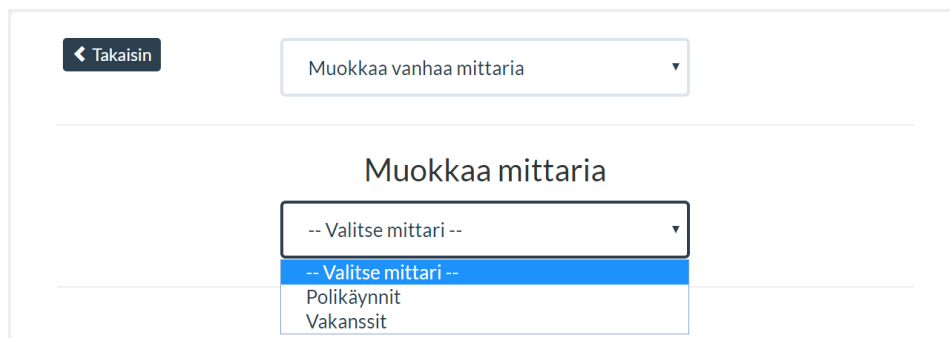
Ali tavoitteiden
 Yli tavoitteiden

Toleranssikohtaan syötät projektin aikana sallitun ylityksen tai alitusprosentti tai yksikköluvun jolloin mittari ei reagoi. Heitto = 1 -kohtaan määrittelet arvot jolla mittarin luku kasvaa yhdellä toleranssin ylityttyä.

Kuva 29. Uuden mittarin luonti

#### 4.9.2 Vanhan mittarin muokkaus

Valitessa vanhan mittarin muokkauksen aukeaa ensin uusi pudotusvalikko, josta valitaan olemassa olevista mittareista se, jota halutaan muokata (Kuva 30).



The screenshot shows a web interface for editing a meter. At the top left, there is a dark button with a left-pointing arrow and the text 'Takaisin'. To its right is a dropdown menu with the text 'Muokkaa vanhaa mittaria' and a downward arrow. Below this is a horizontal line. In the center, the title 'Muokkaa mittaria' is displayed. Below the title is another dropdown menu with the text '-- Valitse mittari --' and a downward arrow. This dropdown menu is open, showing three options: '-- Valitse mittari --' (highlighted in blue), 'Polikäynnit', and 'Vakanssit'. Below the dropdown menu is another horizontal line.

Kuva 30. Valitse muokattava mittari

Kun haluttu mittari on valittu, aukeaa vastaava näkymä kuin uutta mittaria luodessa. Näkymään syötetään ne tiedot, joita halutaan muuttaa ja loput jätetään tyhjäksi. Lisäksi tallennuspainikkeen alla on uusi painike, jolla koko mittarin ja siihen syötetyt tiedot voi poistaa kokonaan. (Kuva 31).

Jätä tyhjäksi kohdat joita et halua muuttaa. Jos muutat lähtöarvoja, aseta kaikki lähtöarvot uudelleen.

<input type="checkbox"/>	Mittarin nimi
<input type="checkbox"/>	Aloituspäivämäärä pp.kk.vvvv
<input type="checkbox"/>	Päätymispäivämäärä pp.kk.vvvv
-- Lähtöarvot ja tavoiteaikajako --	
<input type="checkbox"/>	Lopullinen tehoparannus (%)

Mittari reagoi jos kehitys:

Ali tavoitteiden

Yli tavoitteiden

Toleranssikohtaan syötät projektin aikana sallitun ylitys tai alitus prosentti tai yksikköluvun jolloin mittari ei reagoi. Heitto = 1 -kohtaan määrittelet arvon jolla mittarin luku kasvaa yhdellä toleranssin ylityttyä.

<input type="checkbox"/>	Sallittu toleranssi
<input type="checkbox"/>	Heitto = 1
-- Heiton ja toleranssin tyyppi --	

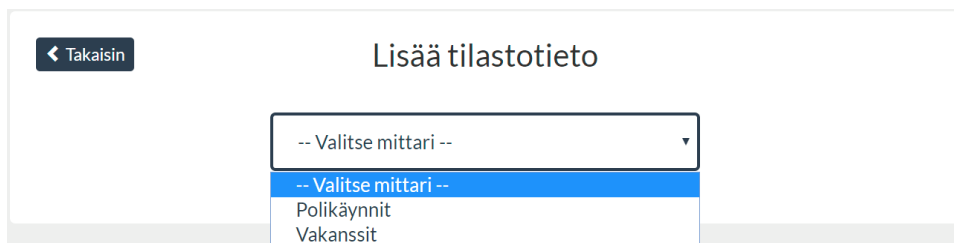
Tallenna muutokset

Poista mittari

Kuva 31. Muokkaa mittaria

#### 4.10 Syöttötyökalu

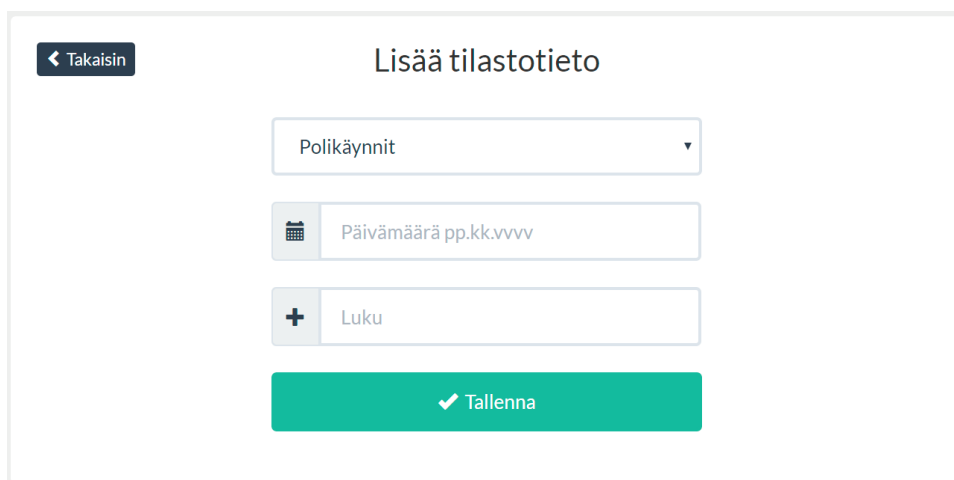
Käyttäjät, joilla on oikeus syöttää mittariin tilastotietoja pääsevät käsiksi tietojen syöttötyökaluun navigointipalkin kautta. Syöttötyökalun avatessa aukeaa ensin pudotusvalikko, josta valitaan haluttu mittari (Kuva 32).



The screenshot shows a web form titled "Lisää tilastotieto". In the top left corner, there is a dark button with a left-pointing arrow and the text "Takaisin". The main content area contains a dropdown menu with the text "-- Valitse mittari --" and a downward arrow. The dropdown is open, showing three options: "-- Valitse mittari --" (highlighted in blue), "Polikäynnit", and "Vakanssit".

Kuva 32. Mittarin valinta, johon tilastoja syötetään

Kun haluttu mittari on valittu, aukeaa uusi näkymä, johon tiedot voi syöttää. Näkymässä on kaksi kenttää: päivämäärä ja luku (Kuva 33).



The screenshot shows the same "Lisää tilastotieto" form. The dropdown menu now shows "Polikäynnit" selected. Below the dropdown are two input fields. The first field has a calendar icon on the left and the text "Päivämäärä pp.kk.vvvv". The second field has a plus sign icon on the left and the text "Luku". At the bottom of the form is a large green button with a white checkmark and the text "Tallenna".

Kuva 33. Syöttötyökalu

Päivämääräkenttä on toteutettu jQueryn datepicker-liitännäisellä. Klikkaamalla päivämääräkenttää aukeaa pieni kalenteri, josta voi valita minkä ajankohdan tietoa mittariin syötetään (Kuva 34). Päivämäärän voi myös kirjoittaa normaalisti kentässä ohjeistetussa muodossa.

The screenshot shows a web interface for adding statistics. At the top left is a button labeled 'Takaisin' with a left-pointing arrow. The main title is 'Lisää tilastotieto'. Below it is a dropdown menu currently showing 'Polikäynnit'. Underneath is a date input field with a calendar icon and the placeholder text 'Päivämäärä pp.kk.vvvv'. A calendar is displayed, showing the month of 'Marraskuu 2017'. The days of the week are abbreviated as 'Ma', 'Ti', 'Ke', 'To', 'Pe', 'La', 'Su'. The date '8' is highlighted in yellow. The calendar grid shows dates from 1 to 30.

Kuva 34. Syöttötyökalun päivämäärän valinta

Jos tiedot on annettu oikeassa muodossa, niin tallennus-painikkeella voidaan tallentaa tieto kantaan, jonka jälkeen tieto on näkyvässä tilastonäkymässä.

## 5 POHDINTA JA YHTEENVETO

Kävin opinnäytetyössäni läpi projektin kehitysvaiheen tavoitteiden seuranta työkalun, eNollan, kehitykseen käytettyjä työkaluja ja menetelmiä sekä itse tuotteen ominaisuudet ja näkymät. Lisäksi esittelin vaihtoehtoisia työkaluja ja menetelmiä, joilla kyseisen tuotteen olisi voinut mahdollisesti toteuttaa. Vertailin myös joitain käytettyjen ja vaihtoehtoisten työkalujen hyviä ja huonoja puolia sekä perustelin syitä, jotka johtivat käytettyjen työkalujen valintaan. Pääosin valinnat perustuivat tilaajan yrityksen käytäntöihin ja omiin tottumuksiini. Olin työsuhteessa tilaajan yrityksessä opinnäytetyötuotetta kehittäessäni.

Lähes kaikki kehityksessä käytetyt työkalut olivat jo jollain tasolla entuudestaan tuttuja itselleni aloittaessani opinnäytetyön tekemistä. Google Charts -palvelu oli itselleni ainoa täysin tuntematon tekijä ennen työsuhteeni alkua tilaajan yrityksessä. Bootstrapin perusteet opiskelin itsenäisesti ennen työsuhteen alkua, koska työpaikkailmoituksessa toivottiin sen osaamista. Muista tekniikoista oli opintojeni aikana käyty vähintään perusteet.

Google Charts -palvelu tuli melko nopeasti tutuksi työsuhteeni aikana jo ennen itse opinnäytetyöhön paneutumista, sillä sitä käytettiin monessa muussakin yrityksen tuotteessa erilaisten kaavioiden luontiin. Opin sen

käytön kuitenkin kattavammin vasta eNollaa kehittäessä, sillä se oli ensimmäinen tuote, jossa pääsin itse alusta asti kokeilemaan erilaisten kaavioiden luontia ja niihin datan yhdistämistä.

Tulevaisuuden kannalta hyödyllisimmäksi kokemukseksi koen web-tekniikoiden yleisen osaamistason kehittymisen. Koin oppivani eniten uutta erityisesti PHP:n, jQuery:n ja Bootstrapin käytöstä yhdessä muiden web-tekniikoiden kanssa dynaamisten, responsiivisten ja relaatiotietokantaohjelmistojen kanssa vuorovaikutuksessa toimivien web-palvelujen kehityksessä.

Listamastani vaihtoehtoisista työkaluista näkisin hyödyllisimpänä joihinkin PHP:n ohjelmointikehyksiin, kuten Laraveliin tai Symphonyyn, tutustumisen. Näkisin, että PHP:n ohjelmointikehykset olisivat tulevaisuudessa hyödyllisiä muun muassa ohjelmointituottavuuden parantamiseen. Olen toisaalta erittäin tyytyväinen, että sain paljon lisää kokemusta perinteisestä puhtaasta PHP:stä.

Tätä opinnäytetyötä varten lukemieni lähteiden perusteella myös kiinnostukseni Pythoniin ja Djangoon kasvoivat kovasti. Kun ottaa huomioon, että sellaiset palvelut kuten Google, YouTube, Instagram ja Spotify ovat ottaneet Pythonin ainakin osittain käyttöönsä, näkisin sen syvemmän opettelun hyödyllisenä tulevaisuutta ajatellen.

Tuotteen kannalta pääsin mielestäni tavoitteeseen hyvin. Päänäkymä on yksinkertainen ja tilastonäkymästä pääsee käsiksi kehityksen historiatietoihin kolmen erilaisen kaavion avulla. Tuotetta kehitettäessä päällimmäisenä tavoitteena oli kuukausikohtaisen seurannan toteuttaminen, joka onnistui hyvin. Tuote kaipaisi tulevaisuuden kannalta kuitenkin lisäominaisuuksien viimeistelyä muiden seuranta-ajanjaksojen osalta. Toisaalta yrityksen tuotteet ovat yleisesti pitkälle asiakkaiden toiveiden mukaan räätälöityjä, mikä on käytännössä tarkoittanut sitä, että lisäominaisuuksia kehitetään asiakkaiden toiveiden mukaan tarpeiden syntyessä.

Läpinäkyvällä tavoitteen asetannalla ja henkilöstön osallistamisella voidaan luoda positiivista muutospainetta toiminnan aitoon kehittämiseen. On mielenkiintoista seurata, millainen potentiaalinen vaikutus tuotteella on investointien kuolettamiseen ja näin myös yhteiskuntaan.

## LÄHTEET

Burnette, E. (2005). *Eclipse IDE Pocket Guide*. Newton: O'Reilly Media.

Coronel, C. & Morris, S. (2016). *Database Systems: Design, Implementation, & Management*. 12. painos. Boston: Cengage Learning.

Emilien, K. & Federico, R. (2015). *MariaDB Essentials*. Birmingham: Packt Publishing.

Facebook Inc. (n.d.). Flux. In Depth Overview. Haettu 15.11.2017 osoitteesta <http://facebook.github.io/flux/docs/in-depth-overview.html#content>

Facebook Inc. (n.d.). Flux. In Depth Overview. Structure and Data Flow. Haettu 15.11.2017 osoitteesta <http://facebook.github.io/flux/docs/in-depth-overview.html#content>

Google Inc. (2017). Google Charts. Quick Start. Haettu 22.11.2017 osoitteesta [https://developers.google.com/chart/interactive/docs/quick\\_start](https://developers.google.com/chart/interactive/docs/quick_start)

Hornostaiev, M. (2017). PHP or Python: Best Language for Server Side Development. Blogijulkaisu 11.8.2017. Haettu 14.11.2017 osoitteesta <https://www.upwork.com/hiring/development/php-or-python-for-server-side-development/>

Hovi, A. (2017). Suomalaisen kehittämä MariaDB lehahti letoon – Google ja Wikipedia käyttävät. *Tivi*. Haettu 28.11.2017 osoitteesta [http://www.tivi.fi/Kaikki\\_uutiset/suomalaisen-kehittama-mariadb-lehahti-lentoon-google-ja-wikipedia-kayttavat-6621106](http://www.tivi.fi/Kaikki_uutiset/suomalaisen-kehittama-mariadb-lehahti-lentoon-google-ja-wikipedia-kayttavat-6621106)

Lutz, M. (2014). *Python Pocket Reference*. 5. painos. Newton: O'Reilly Media.

MariaDB (2017). MariaDB versus MySQL - Compatibility. Drop in replacement for MySQL. Haettu 28.11.2017 osoitteesta <https://mariadb.com/kb/en/library/mariadb-vs-mysql-compatibility/>

McFarland, S. (2014). *JavaScript & jQuery*. 3. painos. Newton: O'Reilly Media.

Oracle Corporation (n.d.). Fast & Smart Code Editing. Haettu 13.11.2017 osoitteesta <https://netbeans.org/features/index.html>

Oracle Corporation (n.d.). PHP Development. Powerful PHP Source Code Editor. Haettu 13.11.2017 osoitteesta <https://netbeans.org/features/php/index.html>

Robbestad, S. (2016). *ReactJS Blueprints*. Birmingham: Packt Publishing.

The PHP Group (2017). What is PHP? Haettu 6.11.2017 osoitteesta <https://secure.php.net/manual/en/intro-what-is.php>

Traversy Media (2017). Laravel From Scratch [Part 1] - Series Introduction. YouTube-video 3.6.2017. Haettu 14.11.2017 osoitteesta <https://www.youtube.com/watch?v=EU7PRmCpx-0>

W3Techs (2017). Historical trends in the usage of server-side programming languages for websites. Haettu 14.11.2017 osoitteesta <https://w3techs.com/technologies/history-overview/programming-language>

W3Techs (2017). Usage statistics and market share of Python for websites. Market position. Haettu 27.11.2017 osoitteesta <https://w3techs.com/technologies/details/pl-python/all/all>

World Wide Technology (2017). Top Ten Most Popular Website Development Using Python and Django Platform. Blogijulkaisu 20.6.2017. Haettu 27.11.2017 osoitteesta <https://worldwebtech.weebly.com/blog/top-ten-most-popular-website-development-using-python-and-django-platform>



**eNolla** 3.0

**FLOWMEDIK**

**Polikäynnit**

Aika: 01.01.2015 - 31.12.2018  
Lopullinen tavoite: 20,00 %  
Mittari reagoi: **All tavoitteen**  
Toleranssi: 5:n prosentin heitto tavoitteesta, jonka jälkeen 1:n prosentin kuukausikohtainen heitto tavoitteesta korottaa lukua yhdellä. (9/2017)

Mittari: Polikäynnit

0

Tilastot

**FLOWMEDIK**

Toteuma

Lisää tilastotietoja

Mittarihallinta

Kirjautu ulos

## eNolla, tilastonäkymä

