

Integration of GPS tracking tool for facility maintenance software

Mikko Pakkanen

Bachelor's thesis

August 2017

School of Technology, Communication and Transport

Degree Programme in Software Engineering

Author(s) Pakkanen, Mikko	Type of publication Bachelor's thesis	Date November 2017
	Number of pages 48	Language of publication: English
		Permission for web publication: x
Title of publication Integration of GPS tracking tool for facility maintenance software		
Degree programme Software Engineering		
Supervisor(s) Väänänen, Olli		
Assigned by Elomatic Process Consulting Ltd		
Description <p>Elomatic Process Consulting Ltd assigned the development of a GPS tracking tool as a module for facility maintenance software called 360°tools. The GPS tracking tool was aimed to be used by the facility personnel to help them with various, repeating maintenance operations.</p> <p>GPS tracking tool was to be one of the many features a customer may choose to use in the 360°tools maintenance system.</p> <p>The primary usage of this tool was aimed to be used with mobile devices and more specifically, with a web browser. The personnel of the facility could see the location of their own device and crucial locations of district heating network on Google Maps web mapping service, in real-time.</p> <p>The personnel of the facility is then able to perform various actions with the tool, e.g. editing, moving or deleting data while carrying out daily, weekly or monthly routines related to their network.</p> <p>The majority of the GPS tracking tool was written using modern web techniques such as JavaScript and HTML 5. The Google Maps application programming interface (API), which played a major role as base of the GPS tracking tool, was written in JavaScript.</p> <p>The project resulted in a functional and stylish module for the existing facility maintenance system.</p>		
Keywords (subjects) Facility, maintenance, software, mobile, web, JavaScript, Google Maps		
Miscellaneous		

Tekijä(t) Pakkanen, Mikko	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Marraskuu 2017
	Sivumäärä 48	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi GPS-paikannuksen integrointi kaukolämpöverkoston kunnossapitojärjestelmään		
Tutkinto-ohjelma Ohjelmistotekniikka		
Työn ohjaaja(t) Olli Väänänen		
Toimeksiantaja(t) Elomatic Process Consulting Oy		
Tiivistelmä <p>Elomatic Process Consulting Oy antoi toimeksiannon, jossa tehtävänä oli toteuttaa GPS-paikannus moduulina jo olemassa olevaan kaukolämpöverkoston kunnossapito-ohjelmistoon nimeltä 360°tools. GPS-paikannusmoduuli suunniteltiin helpottamaan kaukolämpölaitoksen henkilökunnan toimintaa erilaisissa kaukolämpöverkoston kunnossapitoon liittyvissä toimenpiteissä.</p> <p>GPS-paikannusmoduulin oli tarkoitus olla yksi monista ominaisuuksista, joita kaukolämpölaitoksen henkilökunnalla on mahdollisuus valita käytettäväksi 360°tools-ohjelmistossa.</p> <p>Primäärinen toiminta moduulille oli suunniteltu tapahtuvan erilaisilla mobiililaitteilla ja tarkemmin mobiililaitteen verkkoselaimella.</p> <p>GPS-paikannusmoduulilla luotiin kaukolämpöverkoston henkilökunnalle mahdollisuus nähdä reaaliajassa mobiililaitteen sijainti ja kaukolämpöverkoston tärkeät pisteet Google Maps-karttapalvelua hyödyntävässä järjestelmässä.</p> <p>Henkilökunnalla on myös mahdollisuus moduulia käyttämällä suorittaa erilaisia toimenpiteitä kaukolämpöverkostonsa osiin, kuten muokata, siirtää ja poistaa dataa liittyen erilaisiin päivittäisiin, viikottaisiin tai kuukausittaisiin huoltotoimenpiteisiin verkostossa.</p> <p>Suurin osa GPS-paikannusmoduulista kirjoitettiin käyttäen apuna moderneja web-tekniikoita, kuten JavaScriptiä ja HTML5:tä. Google Maps-ohjelmointirajapinta, jonka ympärille moduulin toiminta perustuu, on kirjoitettu JavaScriptillä.</p> <p>Työn tuloksena luotiin funktionaalinen ja tyylikäs moduuli jo olemassa olevaan kaukolämpöverkoston kunnossapitojärjestelmään.</p>		
Avainsanat (asiasanat) Laitos, kunnossapito, huolto, ohjelmisto, web, mobiili, JavaScript, Google Maps		
Muut tiedot		

Contents

Figures	7
Terminology	9
Acknowledgements	12
1 Introduction	13
1.1 Elomatic.....	13
1.2 360°tools.....	13
1.2.1 ePlant360°	14
1.2.2 eMap360°	14
1.2.3 360°tools mobile.....	15
1.3 Assignment	16
2 Service environment and technologies used	16
2.1 Service environment	16
2.2 HTML5	16
2.3 CSS	17
2.4 JavaScript.....	17
2.5 jQuery	18
2.6 PHP	18
2.7 MySQL.....	19
3 Requirement analysis	20
4 Planning	21
5 Work environment	22
6 Development	24
6.1 Initial steps	24
6.2 Converting coordinates	24

6.2.1	Linking converted coordinates with existing network points in the database	25
6.2.2	Creating RESTful routes for accessing the data.....	27
6.3	Class diagram of the GPS tracking tool	29
6.4	Initializing Google Maps	30
6.5	Google Maps marker related problems and marker clustering.....	32
6.6	Adding the pipes.....	33
6.7	Attach InfoWindow-object to the Marker-objects	34
6.8	Implementing the item card.....	35
6.9	Implementing the sidebar	36
6.10	Implementing the navigation to address	37
6.11	Implementing the measuring tool.....	39
6.12	Implementing the drawing and the writing tools	39
7	Tests.....	40
8	Problems	41
8.1	Google Maps API Premium	41
8.2	Converting coordinates	42
9	Further development.....	42
9.1	From mobile to desktop	42
9.2	Using a rectangle shape to select area to send an SMS to clients.....	42
9.3	Visualising the network temperatures with the Google Heatmaps.....	43
10	Conclusion.....	44
	References	45
	Appendices	47
Appendix 1.	Screenshot of ePlant360° main view	47
Appendix 2.	Screenshot of eMap360° main view.....	48

Appendix 3. Screenshot of 360°tools Mobile main view	49
Appendix 4. Screenshot of the GPS tracking tool class diagram	50
Appendix 5. GPS tracking tool integrated to eMap°360, final version at the time of writing	51

Figures

Figure 1. ePlant360° main view.....	14
Figure 2. eMap360° main view.....	15
Figure 3. 360Tools° Mobile main view	15
Figure 4. JavaScript selector for element ID “demo”	17
Figure 5. jQuery selector for paragraph-element	18
Figure 6. Simple PHP-file	19
Figure 7. Relation-link between two fictional database tables	20
Figure 8. Project management view in GitHub-service	21
Figure 9. Notepad++ editor	22
Figure 10. MySQL Workbench.....	23
Figure 11. Chrome Developer Tools.....	23
Figure 12. coordtrans.fgi.fi coordinate conversion tool	25
Figure 13. Relational model between vi_netitems and vi_netitem_coordinates	26
Figure 14. Relational model between vi_lines and vi_line_coordinates	26
Figure 15. Simple REST-route created with Flight.....	27
Figure 16. REST-route for fetching coordinate data from the database	28
Figure 17. JavaScript-script implementing AJAX-call to the REST-route.....	28
Figure 18. JavaScript object containing coordinate data in Chrome DevTools	29
Figure 19. GPS tracking tool class diagram	29
Figure 20. drawMarkers-function of the GPSMapDraw-class handles drawing marker-objects to the map.	31
Figure 21. GPS tracking tool initial version	31
Figure 22. JavaScript object defining different types of customers’ network.....	32
Figure 23. GPS tracking tool using MarkerClusterer library for marker grouping	33
Figure 24. drawLines function of the GPSMapDraw handles drawing polyline-objects to the map.....	34
Figure 25. InfoWindow object showing different kinds of actions user can perform .	35
Figure 26. Item Card implementation showing details of network item.....	36
Figure 27. Two sidebars containing different buttons for different kinds of actions..	36
Figure 28. Popup containing input field for destination location	38
Figure 29. Drawn route to the given address.	38

Figure 30. Measuring tool in action 39

Figure 31. Drawing tools in the action 40

Figure 32. Idea behind selecting clients for SMS sending..... 43

Figure 33. Visualization of the heat values in eMap°360..... 43

Figure 34. Google Heatmap service as seen in Google Maps API documentation 44

Terminology

AJAX

AJAX is an acronym for Asynchronous Javascript And XML. It uses combination of a browser built-in XMLHttpRequest object to request data from a web server and Javascript and HTML DOM (Document Object Model) to display or use the data. AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes, which means that it is possible to update parts of a web page, without reloading the whole page. (AJAX Introduction. 2017)

GPS

GPS is an acronym for Global Positioning System. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver. The system consists of 24 earth orbiting satellites that are eleven thousand nautical miles in space and has six different orbital paths. The satellites are moving and they are doing two complete orbits around the earth in just under 24 hours. (What is GPS?. 2017)

HTML

HTML is an acronym for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML elements are the building blocks of HTML pages. HTML elements are represented by tags. Web browsers do not display the HTML tags, but use them to render the content of the page. (What is HTML? 2017)

CSS

CSS is an acronym for Cascading Style Sheets. CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed. (CSS Tutorial. 2017)

LAMP

LAMP is an acronym for Linux, Apache, MySQL, PHP. It is a stack of open source software that is typically installed together to enable a server to host dynamic websites and web apps. Linux is a operating system, Apache is the web server, MySQL is the database and the dynamic content is processed by PHP. (How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04. 2017)

PHP

PHP is an acronym for “PHP: Hypertext Preprocessor” and it is a server scripting language and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used and free. PHP scripts are executed on the server. PHP files can contain text, HTML, CSS, JavaScript and PHP code. (PHP 5 Introduction. 2017)

REST

REST is an acronym for Representational State Transfer and it relies on stateless, client-server, cacheable communications protocol. Virtually in all cases, the HTTP protocol is used. REST is an architecture style for designing networked applications. RESTful applications use HTTP requests to post data, read data and delete data. REST is not a standard and there will never be W3C recommendation for REST. (Learn REST: A Tutorial. 2017)

W3C

The W3C, or World Wide Web Consortium, is an international consortium where member organizations, a full-time staff, and public work together to develop web standards. The mission statement of the W3C is: *To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the web.* (What is the W3C. 2017)

SQL

SQL is an acronym for Structured Query Language and it is a standard language for storing, manipulating and retrieving data in databases. (Introduction to SQL. 2017)

XML

XML stands for eXtensible Markup Language. XML is a markup language much like HTML and it was designed to store and transport data. XML was designed to be self-descriptive. (Introduction to XML. 2017)

JavaScript

JavaScript is the programming language of HTML and the Web. It is dynamic programming language and can provide dynamic interactivity on websites, when applied to an HTML document.

Git

Git is a version control system designated to handle everything from small to very large projects with speed and efficiency. Git is free and open source. Git is tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for management of source code in software development. (Git. 2017)

Acknowledgements

I would like to say thanks to my colleagues at Elomatic who helped me with the technical details, planning and testing of this project and also Elomatic as a company that gave me this assignment and trusted me to complete it.

1 Introduction

1.1 Elomatic

Engineering design company Elomatic Ltd was founded in 1970 by a Master of Science and industry advisor Mr. Ari Elo. One of the most important steps taken business wise for the young company were in 1974, when it was decided to invest in engineering model technology, which then again was used for designing layout and piping of ships. After that, Elomatic grew rapidly and the company's activities were expanded both geographically and industry wise. (Our story 2017)

In the 1980s, Elomatic started to develop their own 3D computer aided plant design system called *Cadmatic*. *Cadmatic* was based on the know-how gained from using engineering models and it quickly became the most advanced 3D plant and ship design software solution on the market. (Our story 2017)

In the 1990s, Elomatic had already grown and diversified steadily over the years and had become a respected international consulting and engineering partner in developing and improving industrial investments and products. By that time Elomatic had developed expertise within machinery and equipment manufacturing, biotechnology, pharmaceuticals, process industries, starch and potato processing, energy generation and distribution as well as marine and offshore. (Our story 2017)

Elomatic has offices in Finland, Poland, India, China, the Netherlands, Italy, Serbia, Russia and United Arab Emirates. (Elomatic in brief 2017)

1.2 360°tools

360°tools is a software family for maintenance and operation, developed by Elomatic. Web-based 360°tools can be used by any device that has access to the internet and has a web browser installed. 360°tools has two main features which the customers can choose to use; ePlant360° and eMap360°. (360°tools – clarity in maintenance 2017)

1.2.1 ePlant360°

ePlant360° (See Appendix 1 or Figure 1) is a plant and property maintenance and management tool. The idea behind ePlant360° is to link devices, maintenance and device documentation to panoramic images of a plant. ePlant360° was designed to significantly ease maintenance and servicing management.

The user interface of ePlant360° is easy to use, fast to learn and customizable for customers' particular needs. (360°tools – clarity in maintenance 2017)

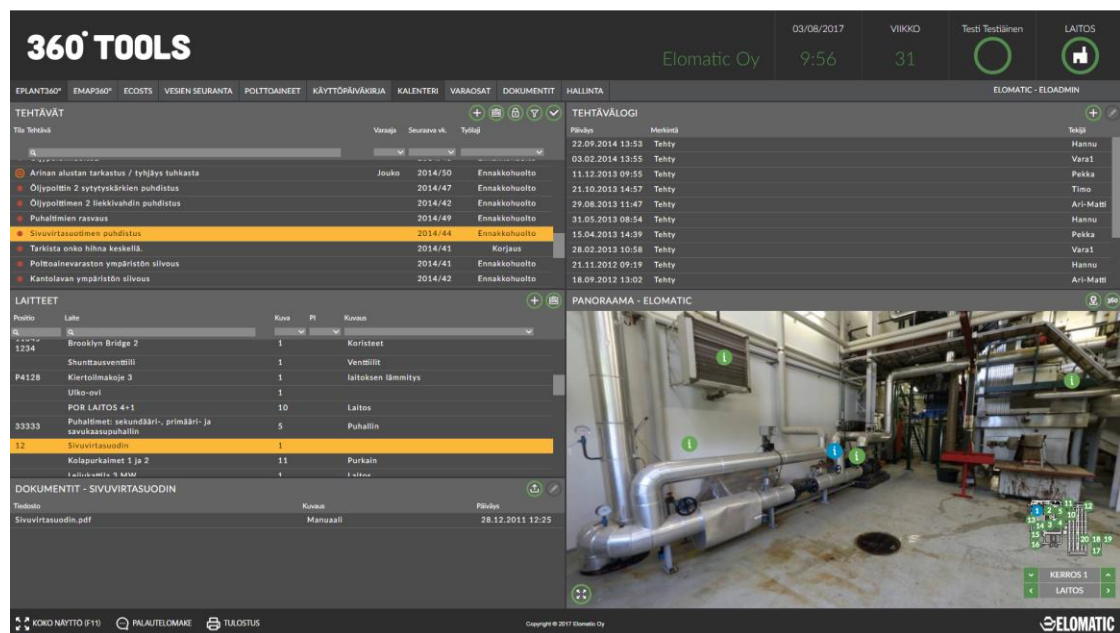


Figure 1. ePlant360° main view

1.2.2 eMap360°

eMap360° (See Appendix 2 or Figure 2) is a network maintenance and management tool. The interface of the eMap360° is very similar to ePlant360° and mostly based on it. Network parts, maintenance information and documentation are linked to a map view.

The user interface of the eMap360° also allows facility personnel to effectively communicate directly with network customers via email and text messages.

(360°tools – clarity in maintenance 2017)

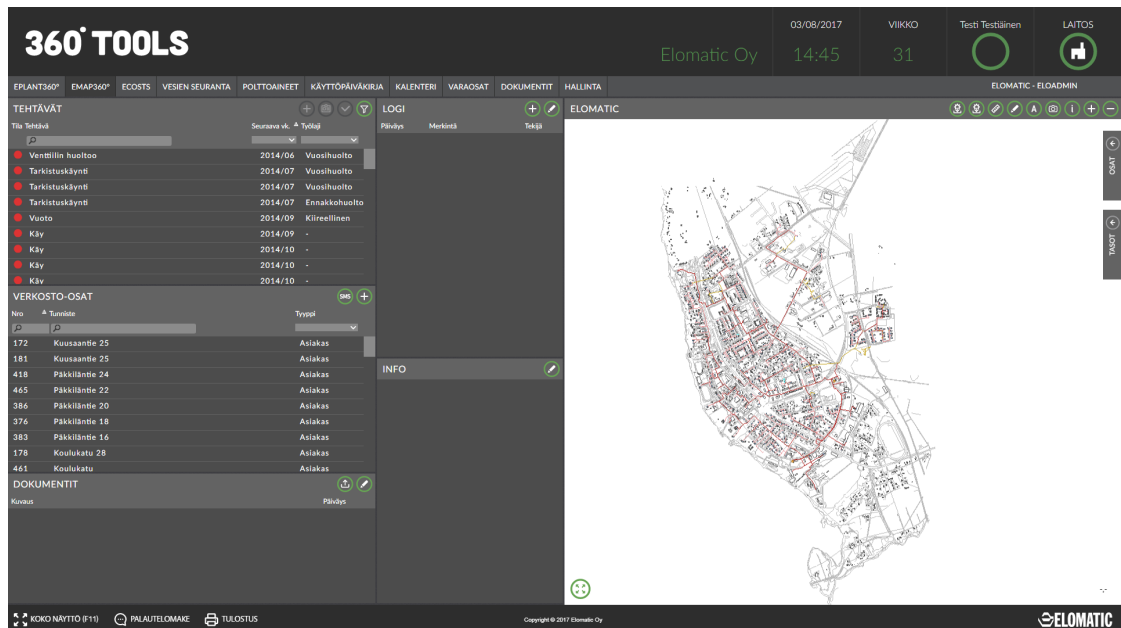


Figure 2. eMap360° main view

1.2.3 360°tools mobile

Both ePlant360° and eMap360° also have a mobile version available (See Appendix 3 or Figure 3). The mobile version allows the facility personnel to do the same operations as they could do with the desktop version. Service is optimized for mobile devices. The items seen in Figure 3 can be customized for each customer.

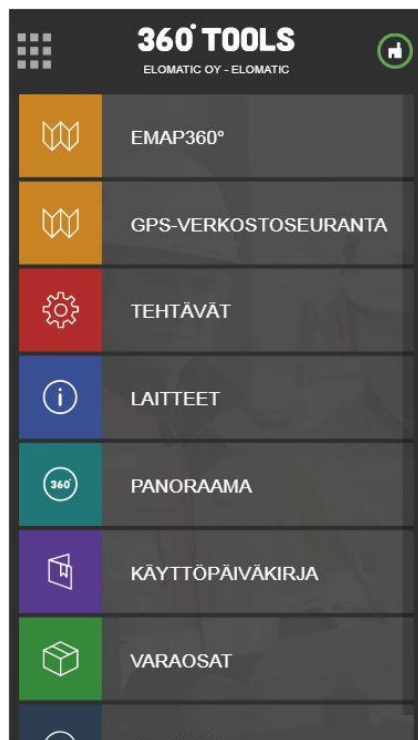


Figure 3. 360Tools° Mobile main view

1.3 Assignment

Elomatic assigned to develop a GPS tracking tool as a module to the mobile version of 360°tools. The basic idea behind the tool was that a facility personnel could use a mobile device, mostly smartphone or tablet, to see their own location on the service based on Google Maps and also crucial locations of their district heating network in real-time while doing routine maintenance operations. They could then add, edit, move or delete data related to the network.

The customers of the 360°tools could then decide whether they wanted to use the module in their set of tools. Attaching the GPS tracking module to 360°tools can be easily done via a control panel, in which the administrators of the system can define customer related properties and perform many administrative actions.

There was no deadline defined for the assignment but most of the hands-on work was accomplished between July 2017 and October 2017. Writing the theoretical part of the project, this thesis, started in August 2017.

2 Service environment and technologies used

2.1 Service environment

360°tools is running LAMP stack (see Terminology, LAMP) as an operation environment. Customer-related MySQL databases are also hosted on the same server. The application itself is accessible with a web browser and requires a user account and a password to login and access services.

GPS tracking tool was primarily developed for the mobile version of 360°tools, although it can also be used with a PC. The initial version and the testing of the tool was conducted by using actual customer data from a production database.

2.2 HTML5

HTML5 is the newest and fifth version of HTML standard and is the standard markup language for creating Web pages. HTML is an acronym for Hyper Text Markup Language. HTML documents consist of HTML elements and HTML elements are

called tags. Tags label pieces of content such as “heading”, “paragraph” or, “table”. Browsers then read these HTML documents as they are requested from the server.

Regardless of the server-side programming language used, HTML will always be formed for the client’s web browser. HTML syntax looks much like XML syntax; it is structured and defines a set of rules for the document. HTML is both human- and machine-readable. (HTML Introduction 2017)

2.3 CSS

CSS is an acronym for Cascading Style Sheets. CSS describes how HTML elements should be displayed on screen, paper or other media. In short, CSS is used to define styles for web pages. Unlike CSS, HTML was never intended to contain tags for formatting a web page. CSS removed the style formatting from the HTML page.

A CSS rule-set consists of a selector and a declaration block. The selector points to the HTML element in which to apply styles. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and value and is separated by a colon. The declaration blocks are surrounded by curly braces. (CSS Tutorial 2017)

2.4 JavaScript

JavaScript is the programming language of HTML and the Web. It should be noted that Java and JavaScript are completely different languages. JavaScript was invented by *Brendan Eich* in 1995 and in 1997 it became an ECMA standard. JavaScript was developed for the Netscape web browser. The first browser to run JavaScript was Netscape 2 in 1996. After Netscape the Mozilla foundation continued to develop JavaScript for the Firefox browser. (JavaScript Tutorial 2017)

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

Figure 4. JavaScript selector for element ID “demo”


2.5 jQuery

jQuery is a popular JavaScript Library. jQuery greatly simplifies JavaScript programming by being lightweight and maintaining the concept of “write less, do more”. jQuery takes plenty of common tasks requiring many lines of regular JavaScript code to accomplish, and wraps them into methods a programmer can call with a single line of code. (jQuery Tutorial 2017)

jQuery also simplifies many complicated things from JavaScript, such as AJAX calls and DOM manipulation. The official jQuery library contains the following features: (jQuery Tutorial 2017)

- AJAX
- CSS manipulation
- HTML event methods
- Effects and animations
- HTML/DOM manipulation

jQuery also has a vast collection of plugins for almost any task. With jQuery one can select HTML elements and perform actions on them. jQuery syntax differs slightly from regular JavaScript; it uses CSS syntax to select elements from HTML document. (jQuery Tutorial 2017)

A screenshot of a code editor showing the jQuery selector `$("p")` inside a light gray rectangular box. The text is in a monospaced font and is highlighted with a light blue background.

```
$("p")
```

Figure 5. jQuery selector for paragraph-element

2.6 PHP

PHP scripts are executed on the server, and PHP is an acronym for “PHP: Hypertext Preprocessor”. PHP is a widely-used, open source scripting language, meaning everyone can freely use it. PHP files can contain text, HTML, CSS, JavaScript and PHP code. PHP codes are executed on the server, and the result is returned to the

browser in the form on an HTML document. PHP files can be recognized by file extension “.php”. (PHP 5 Introduction 2017)

PHP can be used for various actions, e.g. generate dynamic page content, associate with the file system, collect form data, send and receive cookies, modify data in the database, control user-access, and encrypt data. (PHP 5 Introduction 2017)

PHP can be run on various platforms; Windows, Linux, Unix, Mac OS x and many more. It is compatible with almost all servers used today (Apache, IIS, etc.) and supports a wide range of databases. (PHP 5 Introduction 2017)

```
<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>
```

Figure 6. Simple PHP-file

2.7 MySQL

MySQL is a fast, easy-to-use, open-source relational database management system (RDBMS) and it is used by many small and big businesses. MySQL uses a standard form of the well-known SQL, which stands for Structured Query Language. (MySQL – Introduction 2017.)

MySQL works on almost every operating system and is accessible with many languages, such as PHP, perl, C/C++ or Java for instance. MySQL is very quick and works well even with large data sets; it supports large databases, up to 50 million rows or more in a single table. (MySQL – Introduction 2017.)

The default file size limit for a table is 4GB, but it can be increased, if your operating system can handle it. (MySQL – Introduction 2017.)

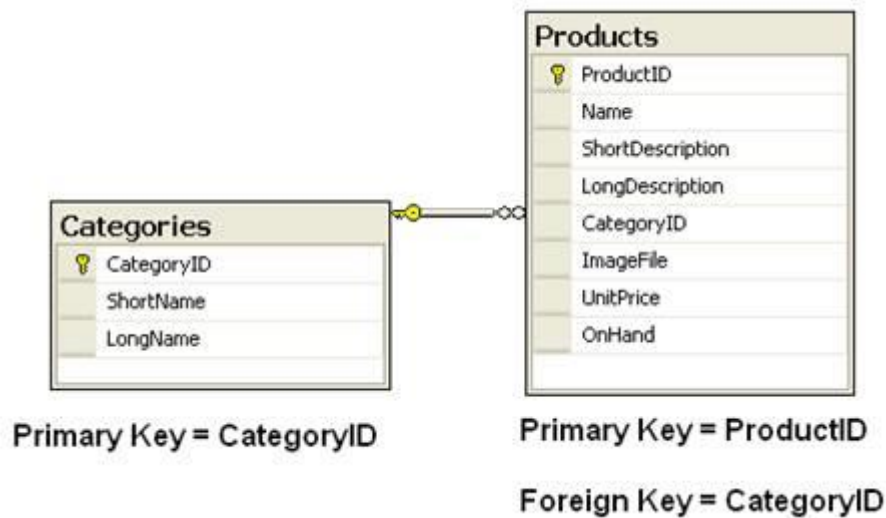


Figure 7. Relation-link between two fictional database tables

3 Requirement analysis

The following requirements were set for the GPS tracking module project:

- It is required of GPS tracking tool module to display crucial points of customers' district heating network, such as clients, wells, facilities, valves and pipes on the map.
- It is required of GPS tracking module to show the user's current location and orientation of the user's device on the map in real time.
- It is required of GPS tracking module to use Google Maps API.
- It is required of GPS tracking tool to be developed as a module which is convenient to deploy into customers' systems using administrative panel in 360°tools.
- It is required of GPS tracking tool to be usable by mobile devices.
- It is required of GPS tracking tool to be developed in a rational and logical manner; approaching the project with the mindset of JavaScript's object-oriented programming.

4 Planning

Initial planning of the GPS tracking module started during the second quarter of 2017. Elomatic already had the idea floating around for building such a tool and during a regular group meeting it was then decided to start a project to develop it, as it turned out one customer of the Elomatic was already showing interest for this kind of tool.

GitHub was used to help with the project planning and management. GitHub is a web-based Git (see terminology: Git) or version control repository and Internet hosting service.

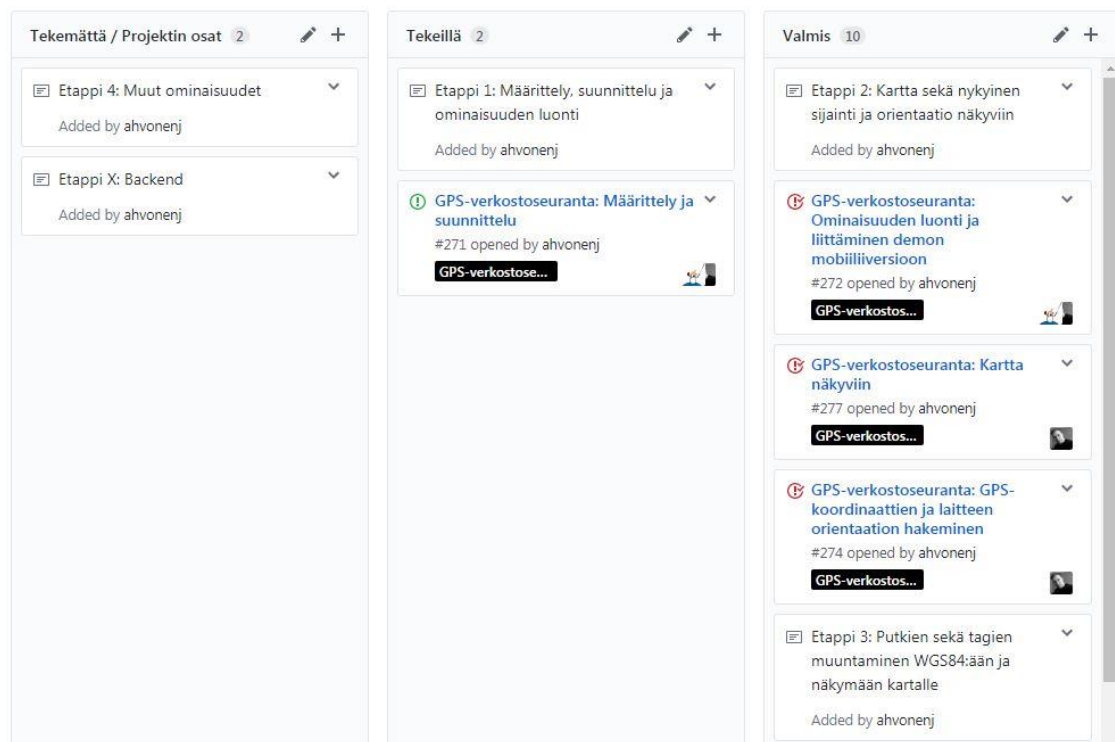


Figure 8. Project management view in GitHub-service

As can be seen from Figure 8, the project had a project management board in GitHub. In this board users with privileges could edit, move and delete cards in each column.

The columns from left to right depict the progress of an individual card or task. On the initial start of the project, the left column contains all the cards, which means those cards are not completed. The center column indicates that the card has work in progress. The cards in the right column are completed tasks.

Each card shown in the board is actually an issue in GitHub. Issues are a great way to keep track of tasks, enhancements and bugs for projects.

5 Work environment

Since the project is completely about web development, simple tools are sufficient. The source code editor Notepad++ was used to write PHP, JavaScript, CSS, HTML and SQL. Notepad++ is a popular source code editor; it has syntax highlighting, code folding, plugins and limited autocompletion as features.

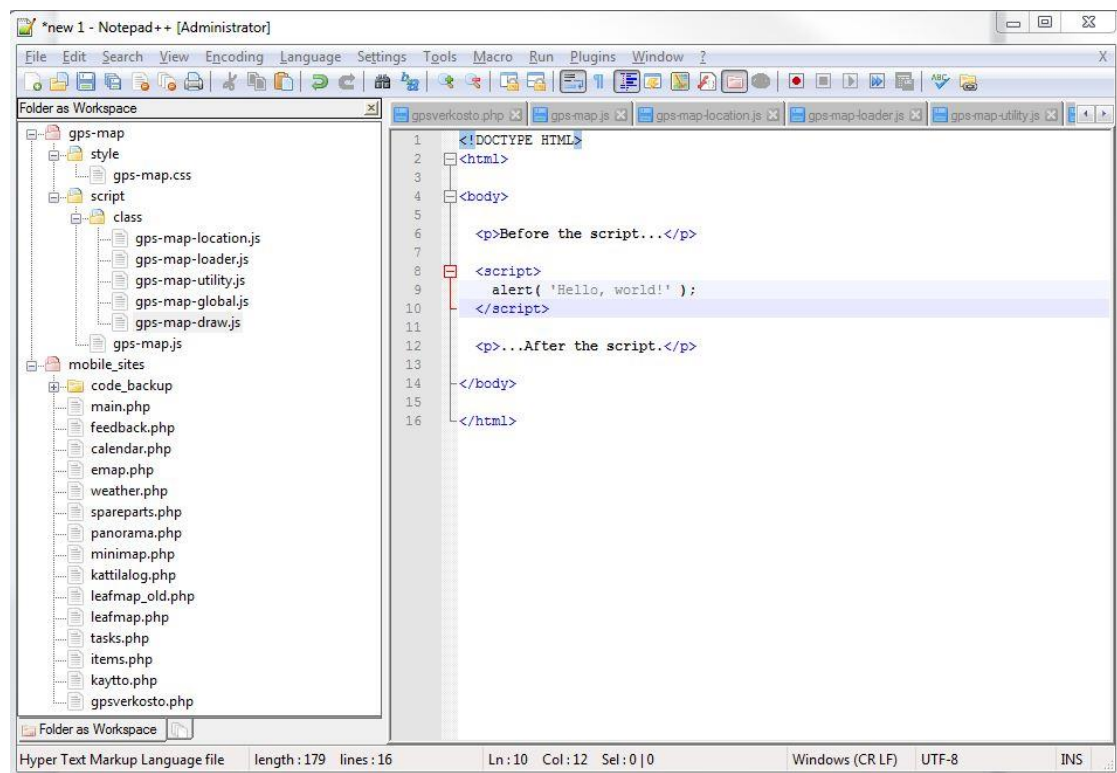


Figure 9. Notepad++ editor

MySQL Workbench was used for database interactions. MySQL Workbench is a visual database design tool developed by Oracle Corporation. It offers SQL development, administration, database design, creation and maintenance in the single integrated development environment for the MySQL database.

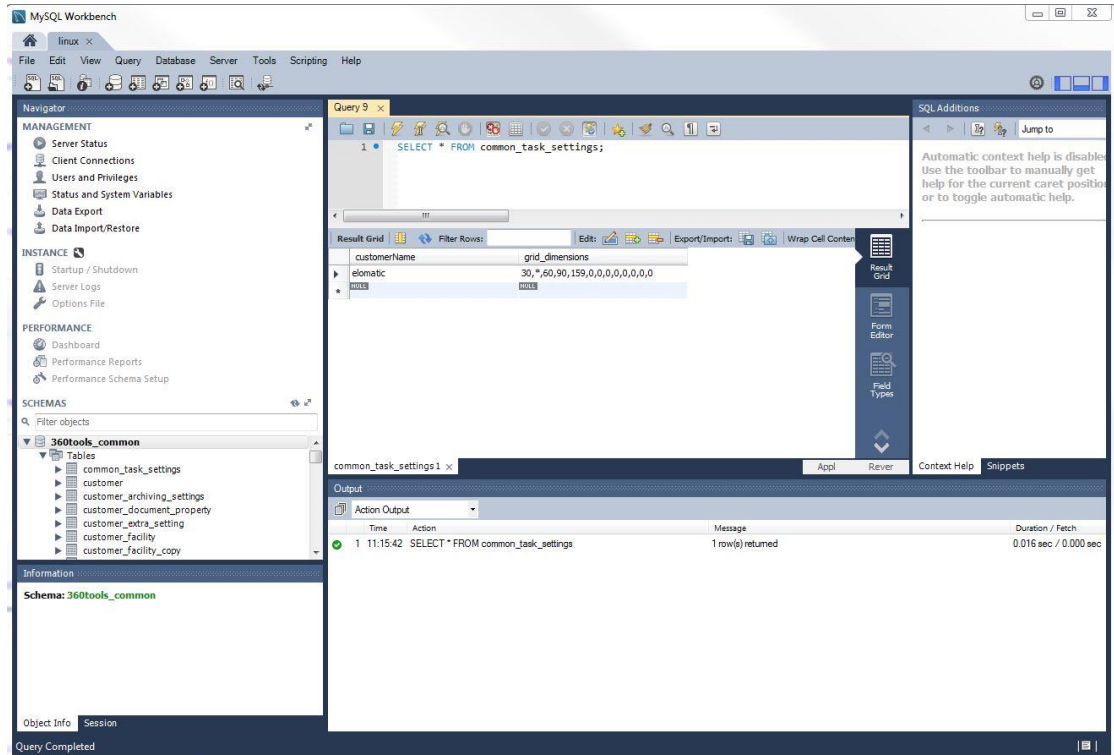


Figure 10. MySQL Workbench

Chrome Developer Tools was used to help in development and debugging. The Chrome Developer Tools (DevTools) are a set of web authoring and debugging tools built in Google Chrome web browser.

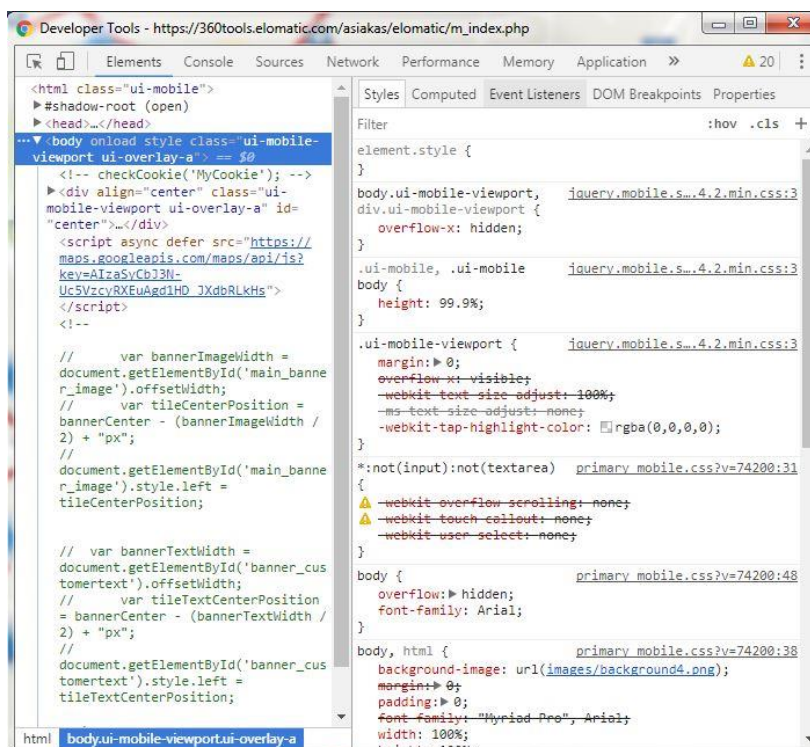


Figure 11. Chrome Developer Tools

6 Development

The development of the GPS tracking module was started in August 2017 after few group meetings and initial planning for what should be implemented.

6.1 Initial steps

It was decided to use Google Maps API as a base of the project. One initial issue with this decision was raised already in the planning phase; Google Maps API was using a different coordinate system than eMap of 360°tools.

Google Maps use *World Geodetic System WGS84*-standard coordinate system, which is almost compatible with coordinate system *ETRS89-EUREF-FIN*, the Finnish realization of *ETRS89* coordinate system.

ETRS89 stands for *European Terrestrial Reference System 1989* and is used in Europe. The difference between coordinate points in *ETRS89-EUREF-FIN* and *WGS84* is less than one meter, hence it is insignificant enough for purposes of this project.

360°tools stores the customers' district heating network coordinates in MySQL database. For testing purposes, one customer database was selected. This particular customer was using the coordinate system *ETRS89-GK25-N2000* for the points of their district heating network. In order to get these points in the right position on the Google Maps would therefore require coordinate conversion from *ETRS89-GK25-N2000* to *ETRS89-EUREF-FIN*.

6.2 Converting coordinates

In order to display the points of the customers' networks on Google Maps, a coordinate system conversion was necessary. For this task, an online, free-to-use conversion tool was used.

Itä-koordinaatti [m]	Pohjois-koordinaatti [m]	Korkeus [m]
25491024.9483	6704279.0282	0
25491031.9035	6704288.5474	0
25491032.4333	6704289.1894	0

Leveysaste	Pituusaste	Korkeus [m]
60.450519238	24.836934502	0.0000
60.450604827	24.837060441	0.0000
60.450610601	24.837070038	0.0000

Figure 12. coordtrans.fgi.fi coordinate conversion tool

The online service, coordtrans.fgi.fi, is able to convert coordinates from one coordinate system to another. The user may input coordinates directly to the left table shown in Figure 12 or input a text-document with coordinate-data to convert them to another system. The conversion system and results are shown in the right table.

In the customers' databases there are two tables in the database containing network data;

- vi_netitems (client, fixed point, facility, pump station, valve, etc.)
- vi_lines (pipes)

vi_netitems-table contains a single item of network per row and vi_lines-table can have many rows per one network item, or in this case, pipe. Both tables also contain coordinate data; however, in the wrong coordinate system. It was required to add two more tables for new, converted coordinate data.

6.2.1 Linking converted coordinates with existing network points in the database

After successfully converting the coordinates from both tables, it was needed to somehow link this new coordinate data with the existing data in the database. The best solution seemed to create two new tables, vi_netitem_coordinates and vi_line_coordinates, for holding new coordinate data and link them together somehow with the two existing tables.

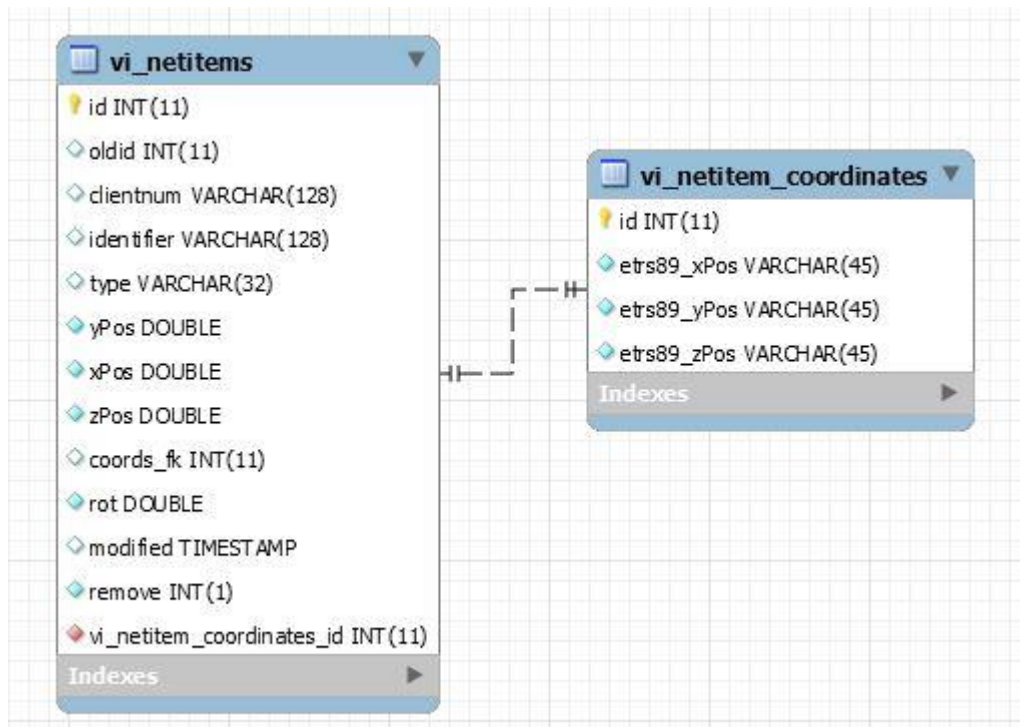


Figure 13. Relational model between *vi_netitems* and *vi_netitem_coordinates*

As can be seen from Figure 13, *vi_netitems*-table has one-to-one relation with *vi_netitem_coordinates*-table, meaning one row from *vi_netitems* is related with one row in *vi_netitem_coordinates*.

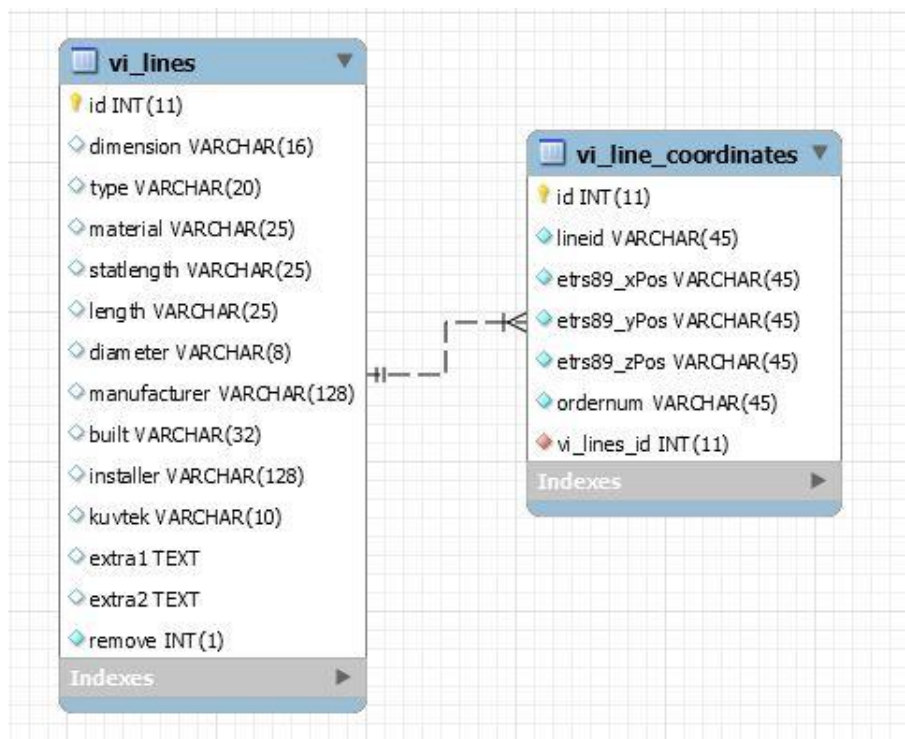


Figure 14. Relational model between *vi_lines* and *vi_line_coordinates*

Figure 14 shows the relation between *vi_lines*-table and *vi_line_coordinates*-table and as can be seen, there is one-to-many relation, meaning one row in *vi_lines* can be related with one or more rows in *vi_line_coordinates*.

6.2.2 Creating RESTful routes for accessing the data

After creating two new database tables and inserting converted coordinate data into them, it was needed to create routes in the application itself for accessing this data. For this, REST (see terminology, REST) routes had to be created.

360°tools uses PHP framework called *Flight* to implement RESTful API. Flight is a fast, simple, extensible framework enabling quick and easy building of the RESTful web applications. Flight requires PHP version 5.3 or greater to work.

```
require 'flight/Flight.php';

Flight::route('/', function(){
    echo 'hello world!';
});

Flight::start();
```

Figure 15. Simple REST-route created with Flight

```

/*
ROUTE: Get ETRS89 coordinates from 'vi_netitem_coordinates'-table
DESC: Returns a list of coordinates for corresponding VerkostoInfo NetItems
PARAMS: facility
SPECIAL: -
*/
function getViCoordinates() {
  if(isset($_GET["facility"]) && !empty($_GET["facility"]))
  {
    $facility = $_GET["facility"];
    $facilityid = Utility::GetCustomerId($facility);
  }
  else
  {
    Utility::Result(true, 100, "Facility not defined");
  }

  $pdo = Utility::GetPDO($facility, "verkostoinfo");

  $query = $pdo->prepare("SELECT VI.coords_fk, VI.identifier, VI.type, VC.etr89_yPos, VC.etr89_xPos, VC.etr89_zPos
                        FROM 'vi_netitems' as VI
                        LEFT JOIN vi_netitem_coordinates as VC
                        ON VC.id = VI.coords_fk
                        WHERE remove = 0
                        ORDER BY VI.id, type ASC");

  $success = $query->execute();

  if($success)
  {
    $result = $query->fetchAll(PDO::FETCH_ASSOC);
    Utility::Result(false, 0, array($result, $facility, $facilityid));
  }
  else
  {
    Utility::Result(true, 600, $customer);
  }
}

```

Figure 16. REST-route for fetching coordinate data from the database

```

/*
Function: AJAX-call for fetching customers VerkostoInfo NetItems data
DESC: Returns a list of customer data, most importantly the coordinates for NetItems for drawing markers to map
PARAMS: -
*/
GPSMapUtility.prototype.getViNetItems = function() {
  var self = this;
  console.log("--- GPSMapUtility fetching VI-NetItems... ---");

  $.ajax({
    type: "GET",
    data: { facility: this.facility },
    url: self.APIURL + "/vi/getvicordinates/",
    success: function (data)
    {
      if(data.error == true) {
        console.log("[GPSMapUtility] Error fetching coordinates for VI-NetItems: \nCode: " + data.code + "\nMsg: " + data.msg);
      } else {
        console.log("--- GPSMapUtility fetching VI-NetItems done! ---");
        //console.log(data);
        self.ViNetItems = data;
        self.viNetItemsFetched.resolve();
      }
    }
  });
}

```

Figure 17. JavaScript-script implementing AJAX-call to the REST-route

As can be seen from Figure 16, there is an SQL-query executed server-side to fetch coordinate data from table *vi_netitems* and *vi_netitem_coordinates* inside *getViCoordinates()*-function, which again is called when accessing a particular URL with HTTP GET request.

Figure 17 then shows an AJAX call (HTTP GET request) made on the client-side to provoke the REST route defined in Figure 16, and thus returning converted coordinate data for later use on the client-side.

```

    ▼ {error: false, code: 0, data: Array(3)}
      code: 0
      ▼ data: Array(3)
        ▼ 0: Array(483)
          ▼ [0 ... 99]
            ▼ 0:
              coords_fk: "1"
              etrs89_xPos: "24.834423710"
              etrs89_yPos: "60.452080950"
              etrs89_zPos: "90.6890"
              identifier: "ILVESKAARI 21"
              type: "Asiakas"
              ▶ __proto__: Object
  
```

Figure 18. JavaScript object containing coordinate data in Chrome DevTools

Two other similar REST routes were also created, one for fetching the pipe coordinates and one for updating the coordinates in *vi_netitem_coordinates* table.

6.3 Class diagram of the GPS tracking tool

At this point it might be a good idea to explain the class structure of GPS tracking tool in more detail. (For a better figure see Appendix 4)

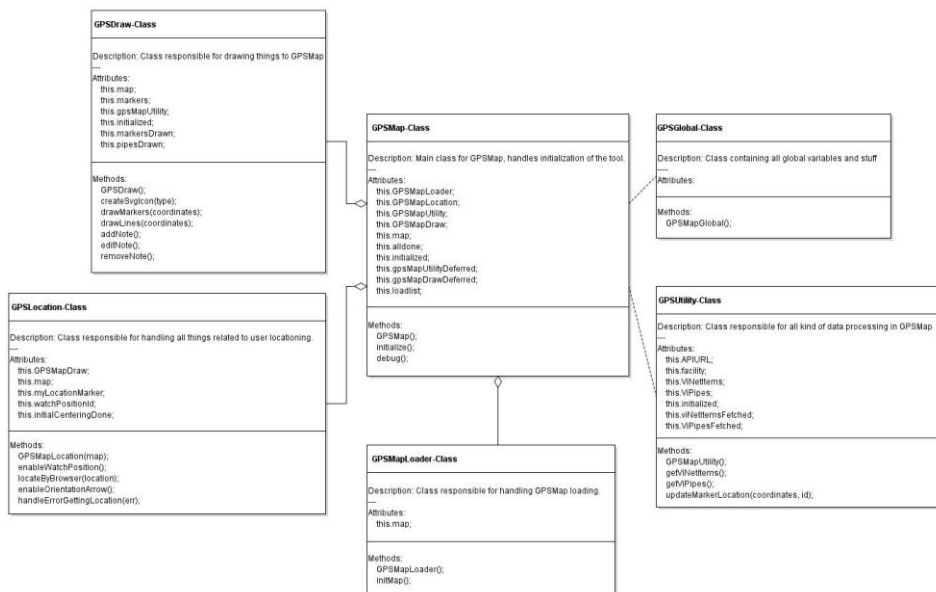


Figure 19. GPS tracking tool class diagram

Figure 19. illustrates the class diagram of the GPS tracking tool. The function of the tool is divided into six different classes; GPSMap, GPSDraw, GPSLocation, GPSMapLoader, GPSMapGlobal, and GPSMapUtility.

Each class has their own responsibility in the big picture:

- GPSMap
 - The main class, handles definition and initialization of any other class.
- GPSDraw
 - Responsible for drawing things to the map.
- GPSLocation
 - Responsible for locating the user and other location related things.
- GPSMapLoader
 - Responsible for loading the map.
- GPSMapGlobal
 - Responsible for holding any global things needed in the tool.
- GPSMapUtility
 - Responsible for fetching data from the database and data processing.

6.4 Initializing Google Maps

After having the REST routes set up and the coordinate data converted and inserted into the database, it was time to initialize Google Maps and try to get some data shown on it.

```

/*
Function: Draw markers to a GPSMap
DESC: Draws markers to a GPSMap from list of coordinates passed as parameter
PARAMS: coordinates
*/
GPSMapDraw.prototype.drawMarkers = function(coordinates)
{
  var self = this;
  if(coordinates) {
    console.log("--- GPSMapDraw drawing markers... ---");

    var coords = {};
    var infowindow;

    // Parse coordinates out of ViNetItems-data
    $.each(coordinates["data"][0], function(index, object) { // Loop through array of objects
      coords[index] = {};
      $.each(object, function(key, value) { // Loop through individual object
        if(key == "etrs89_yPos") {
          coords[index]["yPos"] = value;
        } else if(key == "etrs89_xPos") {
          coords[index]["xPos"] = value;
        } else if(key == "etrs89_zPos") {
          coords[index]["zPos"] = value;
        } else if(key == "type") {
          coords[index]["type"] = value;
        } else if(key == "identifier") {
          coords[index]["identifier"] = value;
        } else if(key == "coords_fk") {
          coords[index]["id"] = value;
        } else if(key == "clientnum") {
          coords[index]["clientnum"] = value;
        }
      });
    });

    // Iterate through coordinates and add markers
    $.each(coords, function(index, object) {
      var myLatLng = new google.maps.LatLng(object.yPos, object.xPos);
      var marker = new google.maps.Marker({
        clickable: true,
        draggable: true,
        icon: self.createSvgIcon(object.type),
        position: myLatLng,
        shadow : null,
        title: object.type + "\n" + object.identifier,
        VI_ID: object.id,
        clientnum: object.clientnum
      });
    });
  }
}

```

Figure 20. drawMarkers-function of the GPSMapDraw-class handles drawing marker-objects to the map.

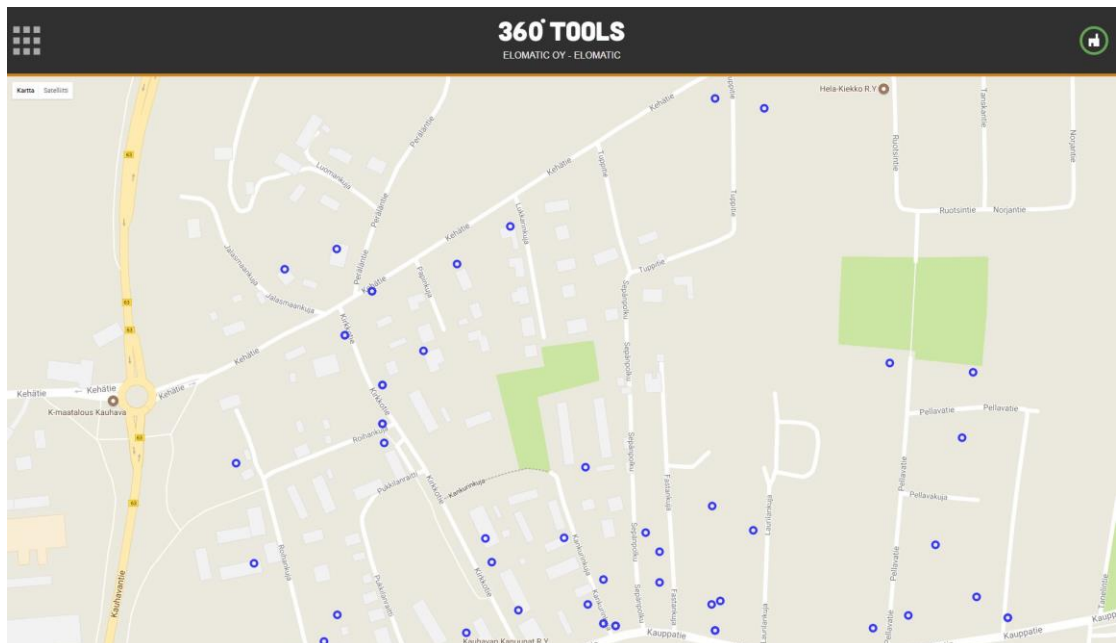


Figure 21. GPS tracking tool initial version

Figure 21 above shows Google Maps initialized and some markers already on the map. One of the first features was to display the correct icon for each marker. As can be seen from Figure 18, the object returned by calling an API route contains property “type”, which is used to segregate different marker types. Figure 21 contains only one type of markers.

```

▼ {MAPTYPES: {...}, TAGTYPES: {...}, worldoffset: null, hastaskiconurl: "h
  ► MAPTYPES: {RAKENNUKSET: {...}, POHJAKARTTA: {...}, KORKEUSKAYRAT: {...},
  ▼ TAGTYPES:
    ► ASIAKAS: {dbid: "PARTS_CLIENTS", name: "Asiakas", tagurl: "https:
    ► KAIVO: {dbid: "PARTS_WELLS", name: "Kaivo", tagurl: "https://360f
    ► KIINTOPISTE: {dbid: "PARTS_REFERENCE", name: "Kiintopiste", tagur
    ► LAITOS: {dbid: "PARTS_PLANTS", name: "Laitos", tagurl: "https://:
    ► LAMPOKESKUS: {dbid: null, name: "Lämpökeskus", tagurl: "https://:
    ► PUMPPU: {dbid: "PARTS_PUMP", name: "Pumppu", tagurl: "https://36f
    ► SUPISTUS: {dbid: "PARTS_REDUCTION", name: "Supistus", tagurl: "ht
    ► VENTTIILI: {dbid: "PARTS_VALVE", name: "Venttiili", tagurl: "htt
    ► VERKOSTOTEHTAVA: {dbid: "PARTS_CLIENTS", name: "Verkostotehtava",

```

Figure 22. JavaScript object defining different types of customers’ network

Figure 22 demonstrates, there are nine different types of customers’ network defined. Each network type has a respective icon in the format of Scalable Vector Graphics (.SVG) file. It was required to use this SVG icon as Google Maps Marker icon.

6.5 Google Maps marker related problems and marker clustering

Apart from having a customized icon for each marker type respectively, there was also another problem; the more markers loaded and shown on the map, the slower the map would get. Customers’ networks typically consist of hundreds of different parts, which in turn means as many markers need to be displayed on the map.

One solution for this was to use a technique called Marker Clustering.

MarkerClusterer is a JavaScript library for creating and managing per-zoom level clusters for large amounts of markers. It will group Google Maps markers to bigger clusters as a user of the map zooms out the view. MarkerClusterer also significantly reduces loading times for the markers on the map, thus making the usage of the map faster. MarkerClusterer objects are green and blue icons shown in Figure 23.

Figure 23 shows a blue arrow as user location on the map, showing user location and orientation in real time.

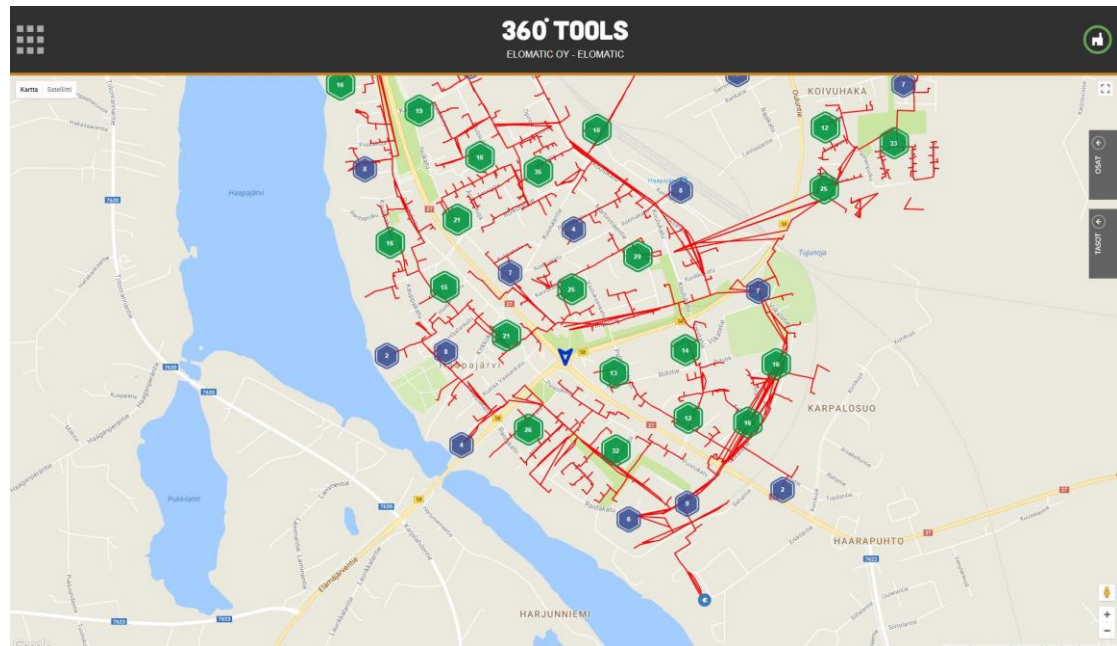


Figure 23. GPS tracking tool using MarkerClusterer library for marker grouping

6.6 Adding the pipes

From Figure 23 can also be depicted that there are red lines showing on the map. These red lines are pipes of customers' network. Pipes are much like markers; they are fetched from the database using AJAX call from the client side to the API, which in turn executes an SQL query to the MySQL server, thus returning the pipe data to the client.

Pipes, however, are not added to the map as Google Maps Marker objects but as Google Maps Polyline objects instead. Polyline objects are used to draw a line on Google Maps. One pipe can consist of many different parts, and the parts must be in a particular order when they are drawn onto the map.

```

/*
Function: Draw pipes to a GPSMap
DESC: Draws VI-pipe to a GPSMap from list of coordinates passed as parameter
PARAMS: coordinates
*/
GPSMapDraw.prototype.drawLines = function(coordinates) {
  var self = this;
  if(coordinates) {
    console.log("--- GPSMapDraw drawing lines... ---");

    var pipes = {};
    var infowindow;

    // Parse coordinates out of VI-pipes data
    $.each(coordinates["data"], function(index, object) { // Loop through array of objects
      pipes[index] = {};
      $.each(object, function(key, value) { // Loop through individual object
        if(key == "diameter") {
          pipes[index]["diameter"] = value;
        } else if(key == "dimension") {
          pipes[index]["dimension"] = value;
        } else if(key == "extra2") {
          pipes[index]["extra2"] = value;
        } else if(key == "id") {
          pipes[index]["id"] = value;
        } else if(key == "type") {
          pipes[index]["type"] = value;
        } else if(key == "coords") {
          var coords = {};
          $.each(value, function(coordsIndex, coordsObject) {
            coords[coordsIndex] = {};
            coords[coordsIndex]["lat"] = coordsObject.x;
            coords[coordsIndex]["lng"] = coordsObject.y;
          });
          pipes[index]["coords"] = coords;
        }
      });
    });

    //console.log(pipes);

    // Iterate through pipes and add polylines (draw pipes to map)
    $.each(pipes, function(index, object) { // Loop through array of objects
      pipes[index] = {};
      $.each(object, function(key, value) { // Loop through individual object
        if(key == "coords") {
          var linePoints = [];
          $.each(value, function(key, value) { // Loop through pipe coordinates object
            linePoints[key] = {};
          });
        }
      });
    });
  }
};

```

Figure 24. drawLines function of the GPSMapDraw handles drawing polyline-objects to the map.

6.7 Attach InfoWindow-object to the Marker-objects

Google Maps also has InfoWindow object used to display content, usually text or images in a popup window above the map at a given location. It was decided to use the InfoWindow object for displaying actions available for user to perform on a particular marker. InfoWindow popup would then open when a user touches or clicks the Marker object. As Figure 25 demonstrates, in the moment of writing there are four different actions for user to perform when clicking or touching the Google Maps Marker object.

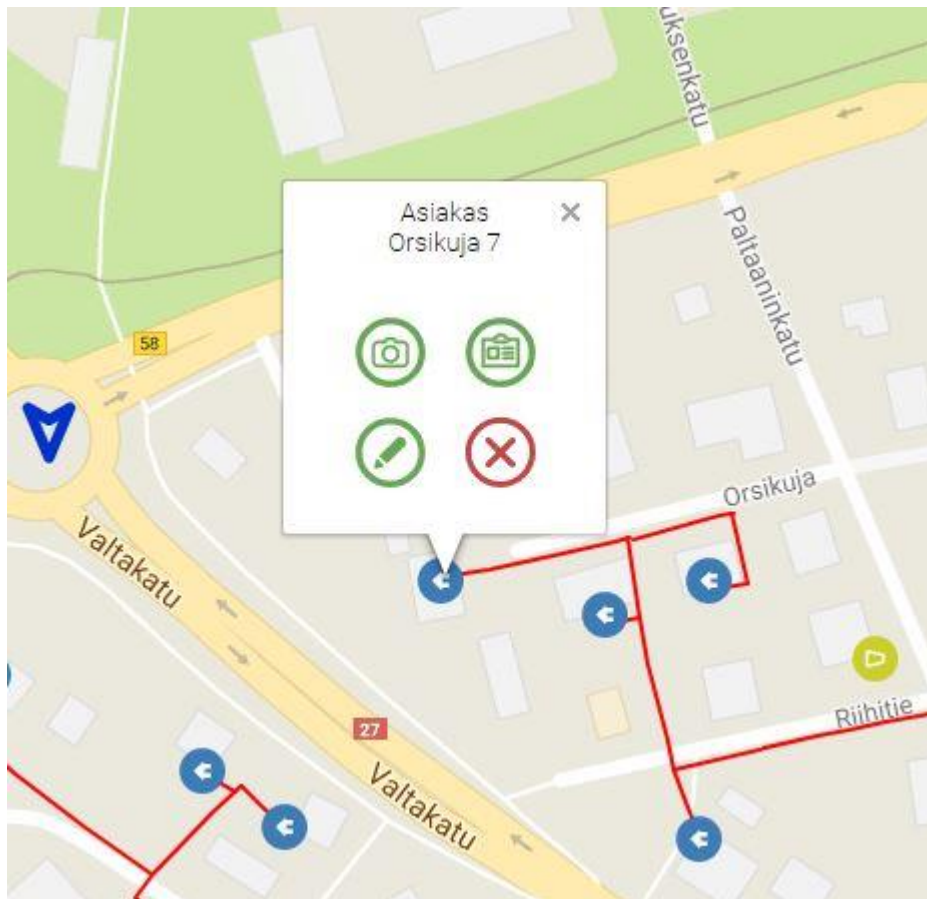


Figure 25. InfoWindow object showing different kinds of actions user can perform

The upper left button is for taking a picture with a mobile device camera and uploading a picture. The picture will be attached to this particular network item. The upper right picture opens an item card containing network details. The bottom left icon also opens the item card for editing, and the bottom right button removes the marker from the map permanently.

6.8 Implementing the item card

It was required to implement an item card for showing details of a network item for the user. When a user clicks on the top left icon of the InfoWindow object, a popup would open and display the data related to the item. (Figure 26).

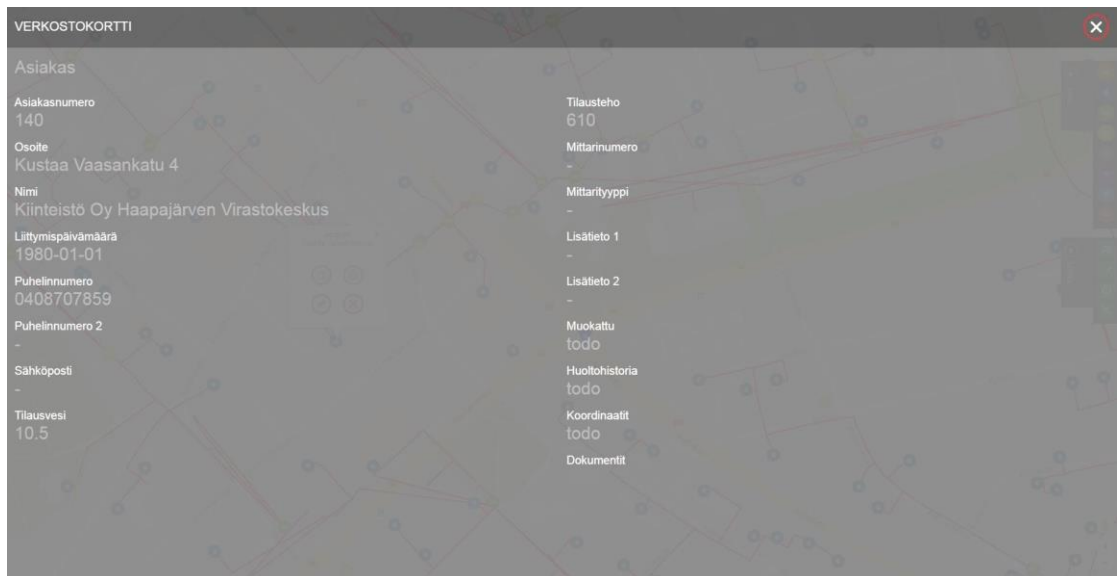


Figure 26. Item Card implementation showing details of network item

6.9 Implementing the sidebar

It was required to implement some convenient menu with which in turn would then perform actions, such as hiding or showing particular markers on the map or opening a popup input field for user to type in an address, which would then be used to draw a route from the user location to the inputted destination. (Figure 27).

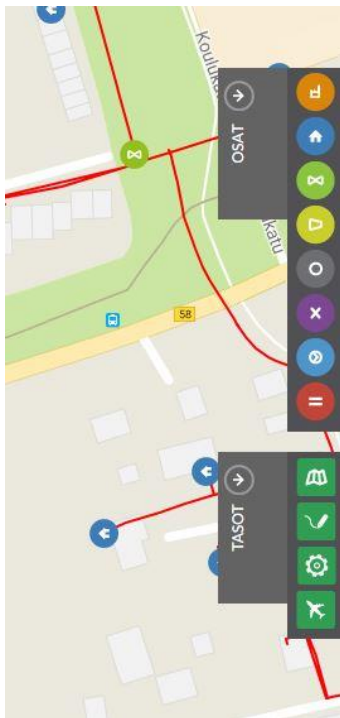


Figure 27. Two sidebars containing different buttons for different kinds of actions

Figure 27 shows two sidebars that were implemented to offer more actions for the user of the map. The uppermost sidebar contains buttons for each marker type on the map. The user may then toggle these buttons on or off and it will hide or show that particular type of marker.

The bottom bar is still under implementation at the time of writing nevertheless the purpose of this toolbar is to hold actions different from just hiding/showing the markers, actions such as drawing a route to the given address, letting users draw their own marks to the map or letting the users define customer-specific settings for the map, et cetera.

6.10 Implementing the navigation to address

It was required for user to be able to input an address and draw the route from the user's current location to the given address. The button for this action is located in the bottom sidebar and by clicking this button, it will open a popup for inputting the address. (Figure 28).

JavaScript library named Vex was used for opening the dialog. Vex is a modern dialog library, which is highly configurable and minimalistic. It has a clear and simple API and works on mobile devices.

As can be seen from Figure 28, the input field will suggest locations when the user starts typing on the text field. For this, a JavaScript library named Geocomplete was used. Geocomplete is an advanced jQuery plugin that wraps Google Maps API's Geocoding and Places Autocomplete services.

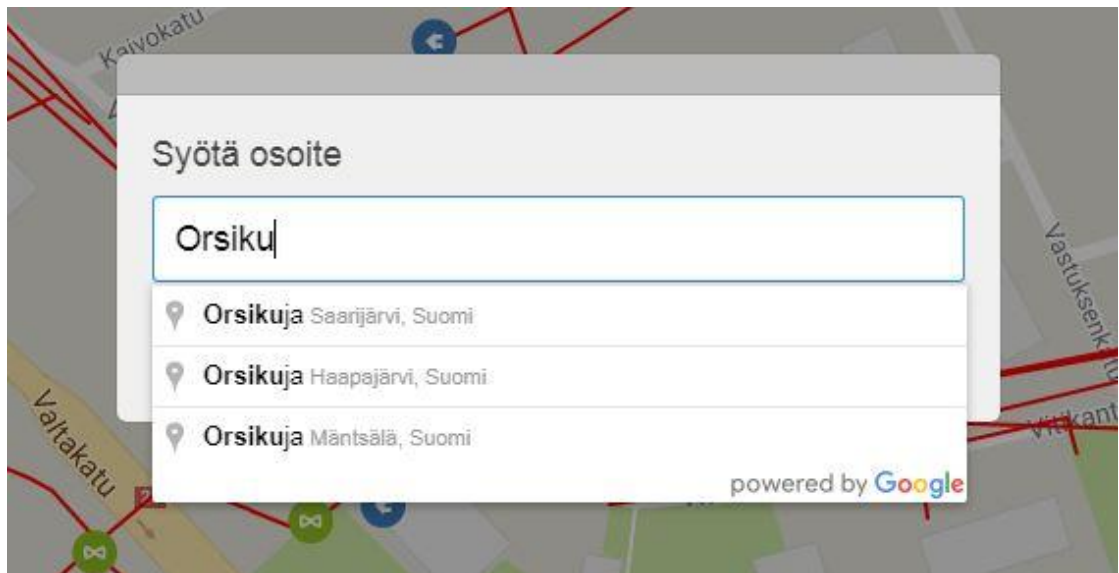


Figure 28. Popup containing input field for destination location

As can be seen from Figure 29, a blue line is drawn on the map as a route to the inputted address.

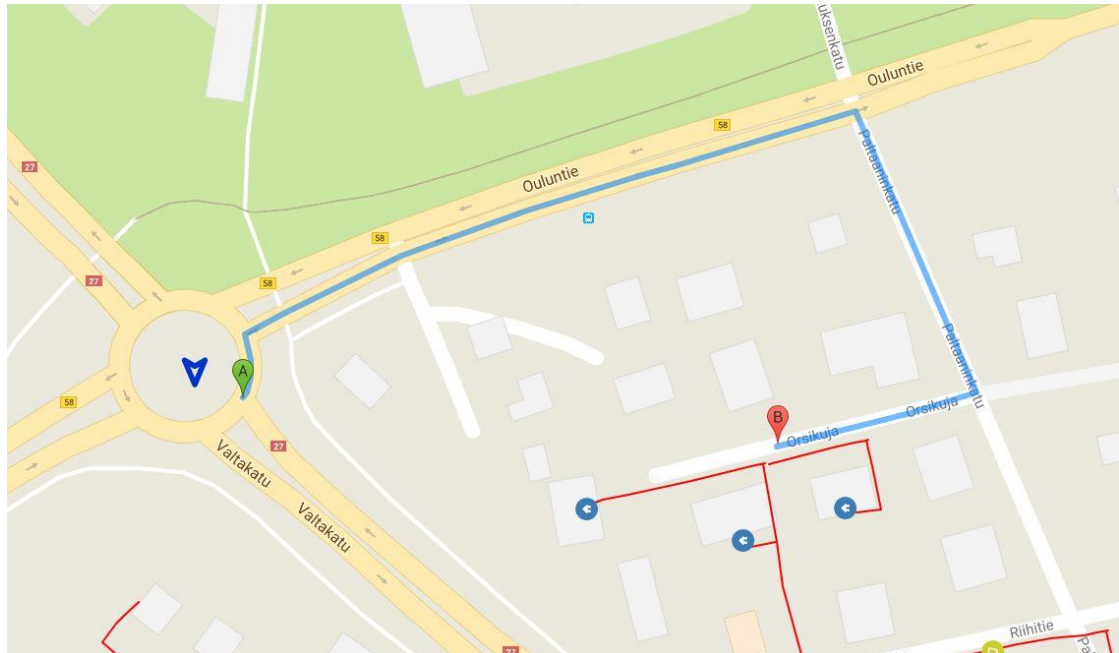


Figure 29. Drawn route to the given address.

6.11 Implementing the measuring tool

It was required to implement a tool for measuring a distance on the map. For this, an external JavaScript library called “MeasureTool-GoogleMaps-V3” was used. The user can activate or deactivate the tool by clicking the ruler-icon in the upper toolbar. After that, the user can start measuring distances by clicking on the map with the left button of the mouse. Tool will then show the distance between two or more points, and can also show total area. (Figure 30).

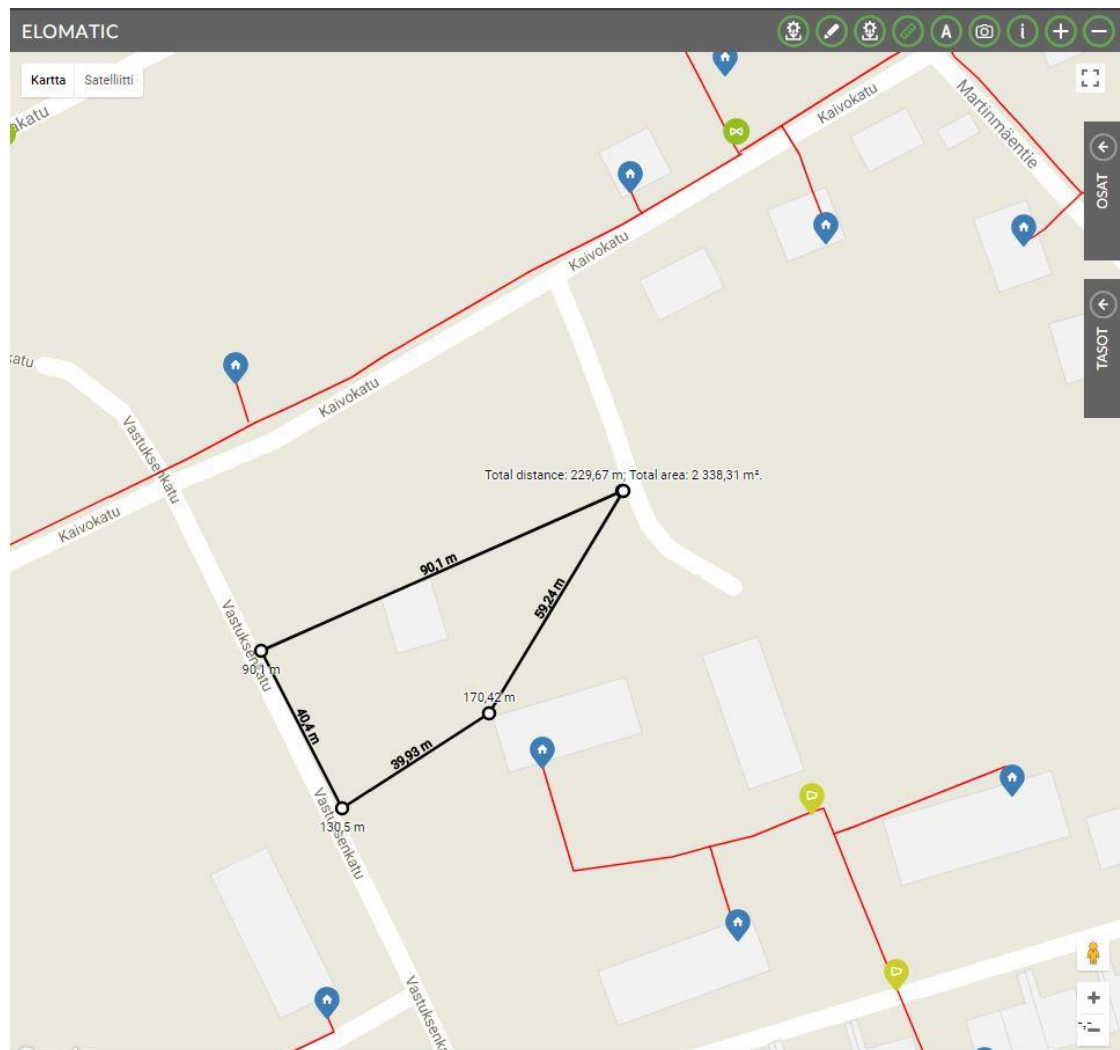


Figure 30. Measuring tool in action

6.12 Implementing the drawing and the writing tools

It was required to implement a tool for drawing shapes of different kinds and also a tool for adding text to the map. To accomplish this, Google Maps Overlays and Drawing Tools libraries were used. The user could activate or deactivate the drawing

tools by clicking the pen-icon in the upper toolbar. Drawing options would then appear and the user could select the desired shape and color to draw it on to the map. The user could also remove drawn shapes from the map by clicking the delete button. (Figure 31).

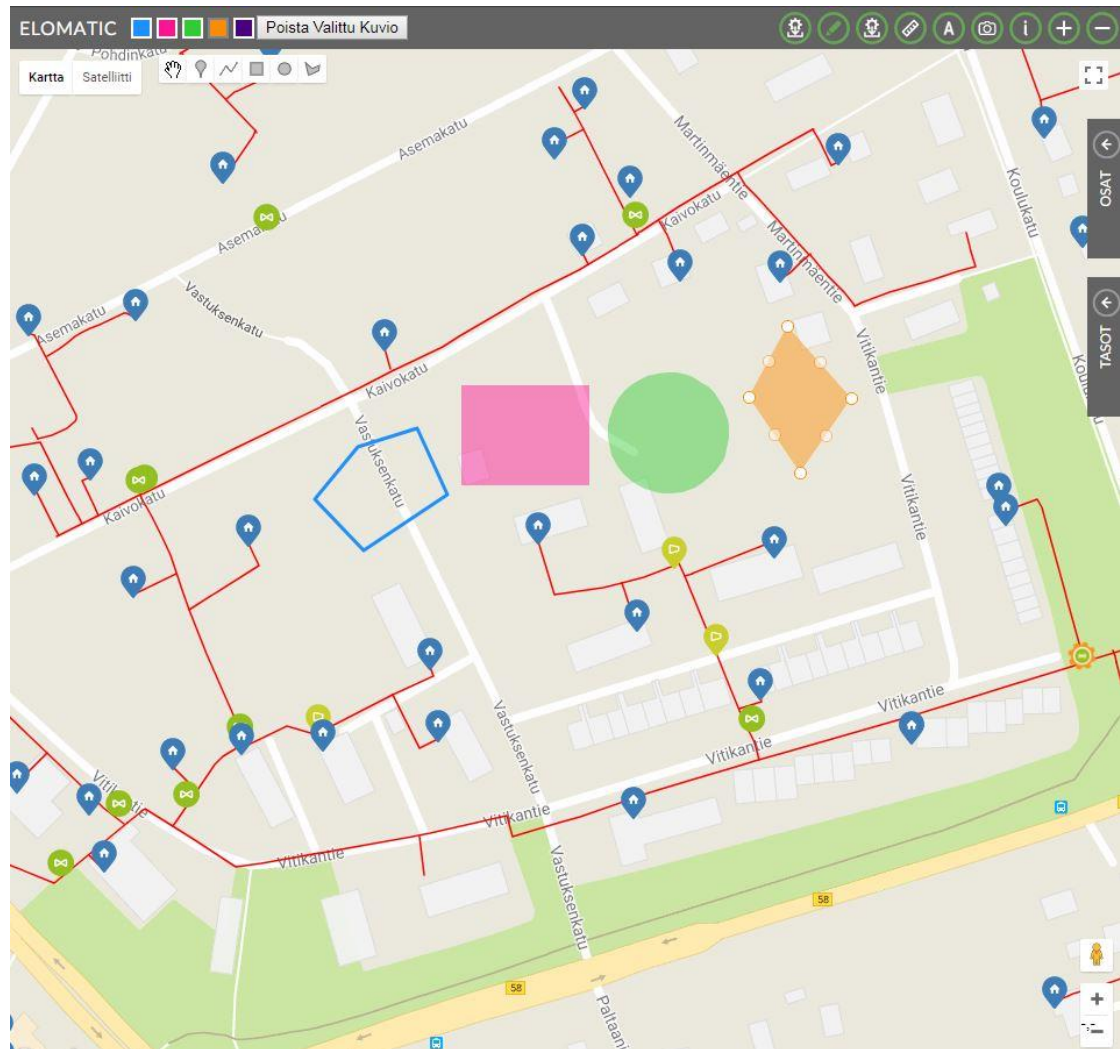


Figure 31. Drawing tools in the action

7 Tests

At the time of the writing there are plenty of frameworks available for testing web software. Unfortunately, in this project there were no time or resources to setup the automated testing system.

However, manual testing played a big part while developing the system and also when integrating the GPS tracking tool into the customer's systems. Practically this

means that testing was done while writing code and fixing the obvious bugs as they appeared.

Manual testing includes steps such as ensuring that everything is working as it should, purposely trying to break the system by inputting arbitrary data to the input fields which are available to the user, clicking and selecting elements and objects, et cetera. Thus, if any bugs were raised during development or during manual testing, they were then fixed. Chrome Developer Tools (See Figure 11) was used a lot for debugging while developing and testing the system.

8 Problems

8.1 Google Maps API Premium

Shortly after the initialization of the project, it was realized that the standard version of the Google Maps API was not sufficient enough for this project, since Google Maps API states the following in their Terms of Service:

Should I purchase the Google Maps APIs Premium Plan?

You should purchase the Google Maps APIs Premium Plan if your business or organization requires any of the following:

- An SLA with 24-hour technical support
- Ability to sell a product or service with embedded Google Maps
- Ability to use Google Maps APIs for an internal application or website
- A specific amount of quota
- Ability to use Google Maps APIs (excluding the Places API) for asset tracking applications
- Premium Plan features available with your existing Maps APIs for Work license

Thus, in this project at least points two, three and five took place. Which means, it was mandatory to purchase the Google Maps API Premium license. Purchasing of the premium license API was discussed in the regular group meeting and it was decided to acquire this license.

8.2 Converting coordinates

Converting the coordinates from old system to the new was one of the initial problems and it had to be solved quickly in order to advance with the project. Conversion of the coordinates is explained earlier in paragraph 6.2 - *Converting coordinates*.

9 Further development

9.1 From mobile to desktop

At the time of writing the GPS tracking system is still in beta-phase, which means that it has been developed and tested using actual customer data but any of the customers are not using it yet. Integration of the tool to the customer's systems will start in November 2017.

On start of the project, this tool was only aimed to be used by mobile devices but this was changed while in development process; the GPS tracking tool appeared to be better than current desktop version of the eMap°360, thus it was agreed to replace the old system with this new tool, which means it was needed to integrate GPS tracking tool to the eMap°360 and ensure that new features introduced in this thesis would work seamlessly with the old features in the eMap°360.

At the time of writing this was already under work and majority of the work was done.

9.2 Using a rectangle shape to select area to send an SMS to clients

One further development idea was to develop a possibility for the user to make a rectangle shaped selection on the map with the mouse. The user could then send an SMS (Short Message Service) message to all of the clients which would be in the selection. (Figure 32).

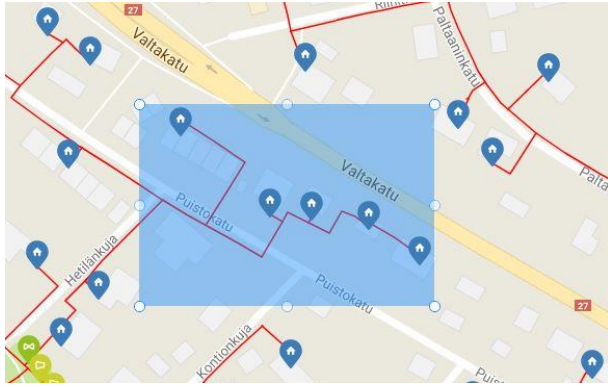


Figure 32. Idea behind selecting clients for SMS sending

9.3 Visualising the network temperatures with the Google Heatmaps

Some of the customers using the eMap³⁶⁰ has an extra feature in use, which visualizes heat values in their network. Each balloon on the map indicates single client's estate and the color of the balloon indicates the heat values in that estate; red means that measured heat from estate is hot and obviously blue stands for cold. (Figure 33).

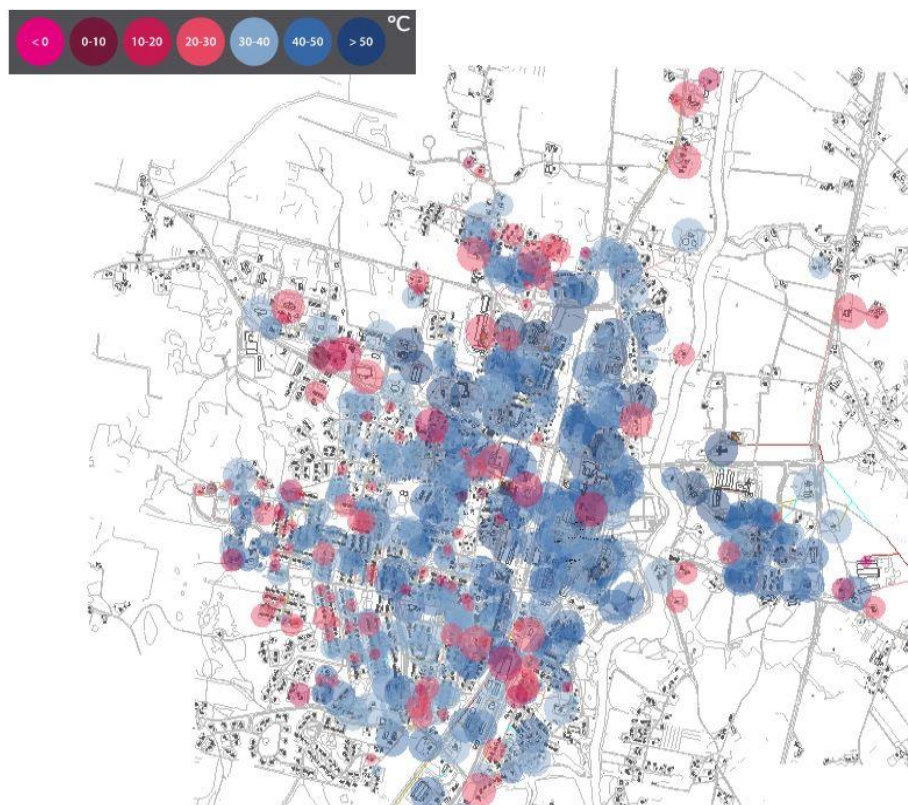


Figure 33. Visualization of the heat values in eMap³⁶⁰

Since the GPS tracking tool is using the Google Maps API and the old map in the eMap°360 is not, it is needed to implement this feature in the Google Maps somehow. For this, the Google Maps API offers service called Heatmap Layer. (Figure 34).

Heatmaps

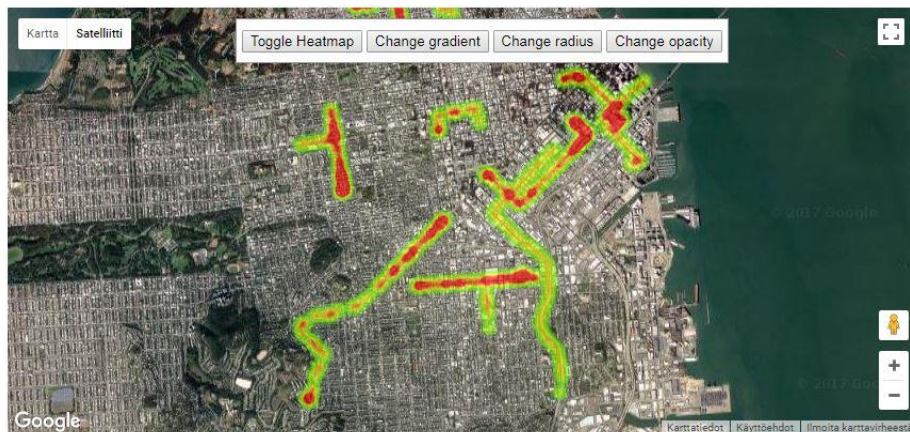


Figure 34. Google Heatmap service as seen in Google Maps API documentation

One of the further development ideas was to replace current visualization system with the one based on Google Heatmaps.

10 Conclusion

The project resulted in a functional and stylish module for the existing facility maintenance system. From my point of view this was really interesting and delightful topic for my thesis. It was entertaining and even challenging at a times to develop this kind of system.

One of the most motivating thing for me was realizing that the developed system got actual interest from the customer side and they would use it to optimize and manage their own business.

It took around four months to develop the system to a point where it had good conditions for the further development and it was good to start integrating it to customer's systems.

References

360°tools – clarity in maintenance. 2017. 360°tools in brief article. Accessed on 01.08.2017. Retrieved from <https://www.elomatic.com/en/products/information-management-software/360-tools/>

AJAX Introduction. 2017. Brief article explaining AJAX. Accessed on 07.08.2017. Retrieved from https://www.w3schools.com/xml/ajax_intro.asp

Bearnes, B. 2016. How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04. Tutorial for installing LAMP. Accessed on 07.08.2017. Retrieved from <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>

CSS Tutorial. 2017. Brief tutorial of CSS. Accessed on 07.08.2017. Retrieved from <https://www.w3schools.com/css/>

Elkstein, M. 2017. Learn REST: A Tutorial. Brief article explaining REST. Accessed on 07.08.2017. Retrieved from <http://rest.elkstein.org/>

Elomatic in brief. 2017. Brief article at Elomatic website. Accessed on 02.08.2017. Retrieved from <https://www.elomatic.com/en/company/elomatic-in-brief.html>

Git. 2017. Git version control system homepage. Accessed on 14.08.2017. Retrieved from <https://git-scm.com/>

HTML Introduction. 2017. Brief article explaining HTML. Accessed on 07.08.2017. Retrieved from https://www.w3schools.com/html/html_intro.asp

Introduction to SQL. 2017. Brief article explaining SQL. Accessed on 07.08.2017. Retrieved from https://www.w3schools.com/sql/sql_intro.asp

Introduction to XML. 2017. Brief article explaining XML. Accessed on 07.08.2017. Retrieved from https://www.w3schools.com/xml/xml_what_is.asp

JavaScript Tutorial. 2017. Quick tutorial for JavaScript. Accessed on 14.08.2017. Retrieved from <https://www.w3schools.com/js/default.asp>

jQuery Tutorial. 2017. Quick tutorial for jQuery. Accessed on 14.08.2017. Retrieved from <https://www.w3schools.com/jquery/>

MySQL – Introduction. 2017. Tutorial introducing MySQL. Accessed on 14.08.2017. Retrieved from <https://www.tutorialspoint.com/mysql/mysql-introduction.htm>

Our Story. 2017. Elomatic in brief article. Accessed on 01.08.2017. Retrieved from <https://www.elomatic.com/en/company/our-story.html>

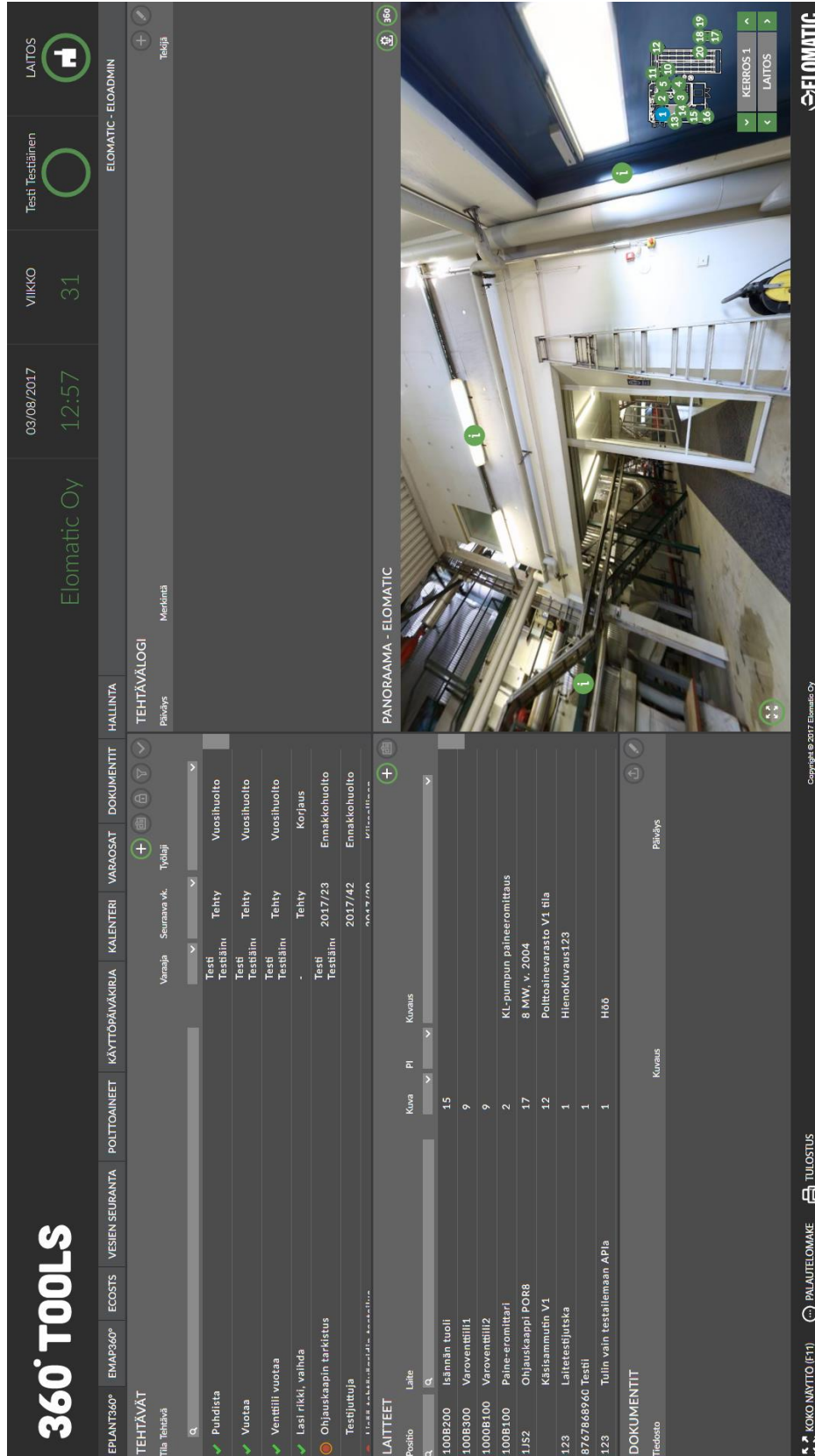
PHP 5 Introduction. 2017. Brief article explaining PHP. Accessed on 07.08.2017. Retrieved from https://www.w3schools.com/php/php_intro.asp

What is GPS? 2017. Article explaining GPS. Accessed on 07.08.2017. Retrieved from <https://www.ja-gps.com.au/what-is-gps.aspx>

What is the W3C. 2017. W3C explained in brief article. Accessed on 07.08.2017. Retrieved from <http://whatisw3c.blogspot.fi/2008/02/w3c-explained.html>

Appendices

Appendix 1. Screenshot of ePlant360° main view



Appendix 2. Screenshot of eMap360° main view

360° TOOLS

Elomatic Oy 14:45 31

03/08/2017 VIKKO Testi Testiäinen LAITOS

ELOMATIC - ELOADMIN

ELOMATIC

TEHTÄVÄT

Tila: Tehtävä

Seuraava vu. Tyyläjä

2014/06	Vuoshiuolto
2014/07	Vuoshiuolto
2014/07	Vuoshiuolto
2014/07	Ennakkohuolto
2014/09	Kiireellinen
2014/09	-
2014/10	-
2014/10	-

VERKOSTO-OSAT

Nro	Tunniste	Tyyppi
172	Kuusaantie 25	Asiakas
181	Kuusaantie 25	Asiakas
418	Päikkiläntie 24	Asiakas
465	Päikkiläntie 22	Asiakas
386	Päikkiläntie 20	Asiakas
376	Päikkiläntie 18	Asiakas
383	Päikkiläntie 16	Asiakas
178	Koulukatu 28	Asiakas
461	Koulukatu	Asiakas

LOGI

Päiväys

Merkintä

TEKIJÄ

KÄYTTÖRÄVÄKIRJA

KALENTERI

VARAOSAT

DOKUMENTIT

HALLINTA

ELOMATIC

INFO

DOKUMENTIT

Kuvaus

Päiväys

KOKO NÄYTTÖ (F11)

PALAUTELOMAKE

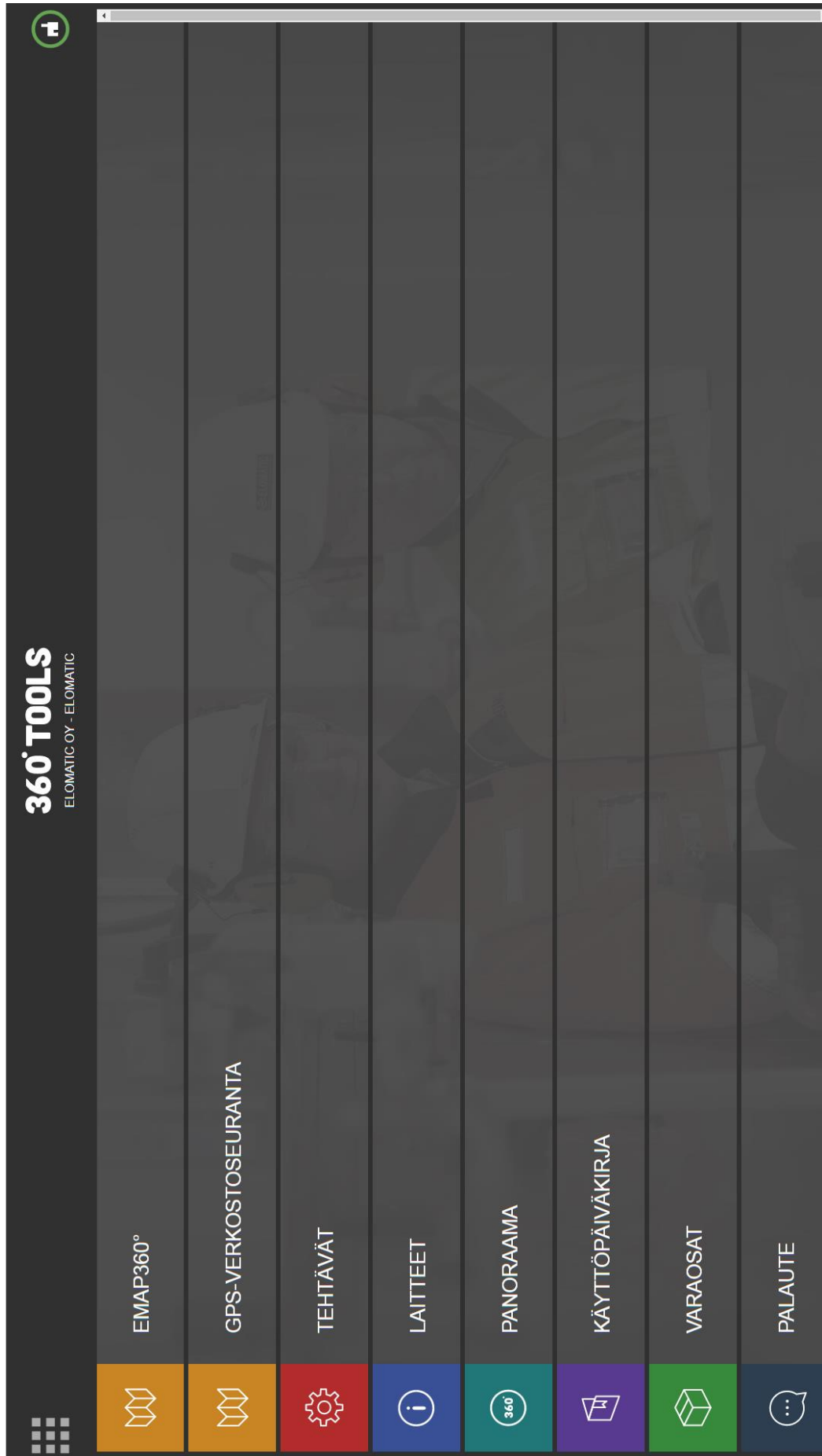
TUULOSTUS

Copyright © 2017 Elomatic Oy

ELOMATIC

Appendix 3.

Screenshot of 360°tools Mobile main view



Appendix 4. Screenshot of the GPS tracking tool class diagram

