

Saimaa University of Applied Sciences  
Faculty of Technology, Lappeenranta  
Double Degree  
Information Technology

Vojtěch Cima, Dalibor Vaníček

## **Outotec Follow-up Form Application**

Thesis 2013

## **Abstract**

Vojtěch Cima, Dalibor Vaníček

Follow-up Form Application, 38 pages

Saimaa University of Applied Sciences

Faculty of Technology, Lappeenranta

Double Degree

Information Technology

Thesis 2013

Instructors: Lecturer Yrjö Utti, Saimaa University of Applied Sciences

Eero Enovaara, Senior Specialist - CRM applications, Outotec

The main purpose of this thesis was to suggest and create an electronic version of follow-up form for Outotec Company. Another goal of this work was to get familiar with new technology features and new development processes of a real company. Information and knowledge was gathered mostly from the internet resources.

The created web application (follow-up form) should be compatible with an Apple iPad and should enable to use the form also without internet connection. State of the art technology, such as HTML5, CSS3, JavaScript and PHP were used during development process.

The final result of this thesis is a web application, which has met all requirements. This application enables to collect, edit and store customer's data without using a paper version of the mentioned form.

The work was consulted and supervised by Mr. Eero Enovaara. Source data for development and the origin paper form was provided by Outotec Company as well.

Keywords: HTML, CSS, JavaScript, PHP

## **List of terms**

AJAX – Asynchronous JavaScript and XML

CERN - The European Organization for Nuclear Research

CRM – Customer Relationship Management

CSS – Cascading Style Sheets

CSV – Comma-Separated Values

DOM – Document Object Model

DROPBOX – File hosting service

FTP – File Transfer Protocol

HTML – HyperText Markup language

HTTP – HyperText Transfer Protocol

JQUERY – JavaScript library

JSON – JavaScript Object Notation

MIME – Multipurpose Internet Mail Extensions

PHP – PHP: Hypertext Preprocessor, Server side scripting language

SGML – Standard Generalized Markup language

Tablet – One piece mobile computer

TAG – HTML basic element

TCP – Transmission Control Protocol

## Table of contents:

1. INTRODUCTION.....	1
1.1. OUTOTEC COMPANY.....	1
2. ANALYSIS.....	3
2.1. CURRENT SYSTEM.....	3
2.2. REQUIRED SYSTEM.....	5
3. DEVELOPMENT.....	7
3.1. DEVELOPMENT TOOLS.....	7
3.2. TECHNOLOGIES.....	8
3.2.1. HTML.....	8
3.2.2. CSS.....	11
3.2.3. PHP.....	12
3.2.4. JavaScript.....	13
3.3. DEVELOPMENT PROCESS.....	17
4. FOLLOW-UP FORM APPLICATION.....	18
4.1. FORM DESCRIPTION.....	18
4.2. FEATURES.....	21
4.2.1. Data saving procedure and client-side local storage (app cache) .	21
4.2.2. Form field auto complete feature.....	24
4.2.3. Editing stored records.....	26
4.2.4. Dynamic form layout feature.....	27
4.2.5. Tradeshow name setup.....	29
4.3. DEPLOYMENT PACKAGE DESCRIPTION.....	30
4.3.1. businessareas.txt file specification.....	30
4.3.2. companies.txt file specification.....	32
4.3.3. data.txt file specification.....	32
5. CONCLUSION.....	34
REFERENCES.....	35
LIST OF FIGURES.....	36
LIST OF SOURCE CODES.....	37
LIST OF TABLES.....	38

## **1. Introduction**

Development of smart portable devices and tablets brings more and more common activities such as consuming web content, reading books, watching movies or listening music right to these devices. Tablets also enable to enter data in a really comfortable way.

In these days applications widely replace a lot of office activities. Gathering and storing data are the two most important purposes for forms itself. Replacing a paper version of form by a web application provides several advantages led by high user comfort during processing.

A really important part of every development is to choose a suitable development process gathering all requirements, correctly analyzing and properly implementing them are the basic pillar for creation of a successful final product.

Current technology provides possibilities to create a requested form in full scale of requirements. The Final application is a deployment package containing files based on HTML5 (HyperText Markup Language), CSS3 (Cascade Style Sheets), JavaScript and PHP (PHP: Hypertext preprocessor) technology. The Final application should be implemented to CRM (Customer Relationship Management) system of Outotec Company and used during upcoming tradeshows for gathering and storing data.

### **1.1. Outotec Company**

Outotec is a worldwide company providing solid-liquid separation solutions. This company has been keeping a strong position across the whole manufacturing chain from mine processing to metal manufacture and currently employs more than 3000 employees in the whole World (1).

Currently Outotec Company provides solutions, development and technology mainly for the following industry areas:

- Minerals and metals

- Chemical
- Energy
- Water treatment

Sophisticated global network of sales and service centers combined with using environment friendly technologies and manufacturing processes helps to keep high profitability and strong market position. Continuous innovations such as automation and new information management (information technology solutions, information systems, et cetera), combined with state-of-the-art research facilities destines this corporation to potential growth in future and getting even larger market share.

The basic division of Outotec business areas:

1. Non-ferrous solutions

This area includes all solutions for processing copper, nickel, zinc, lead, gold, silver and PGM's from mine to pure metal. It also contains research and development facilities dealing with this area.

2. Ferrous solutions

This area covers all solutions for processing iron ore, ferroalloys and titanium feedstock.

3. Energy, Light Metals and Environmental Solutions

These are solutions for new applications of proven technologies.

Outotec Company also provides after sale service based on unique approach of combination providing sustainable solutions, products and related customer service. Figure 1 shows Outotec Company building in Lappeenranta (Finland).



Figure 1 - Outotec Company building in Lappeenranta

## 2. Analysis

This chapter contains description of the current system, the paper version of the follow-up form. It also includes description of the required application. The main idea of system transformation is described in Figure 2.

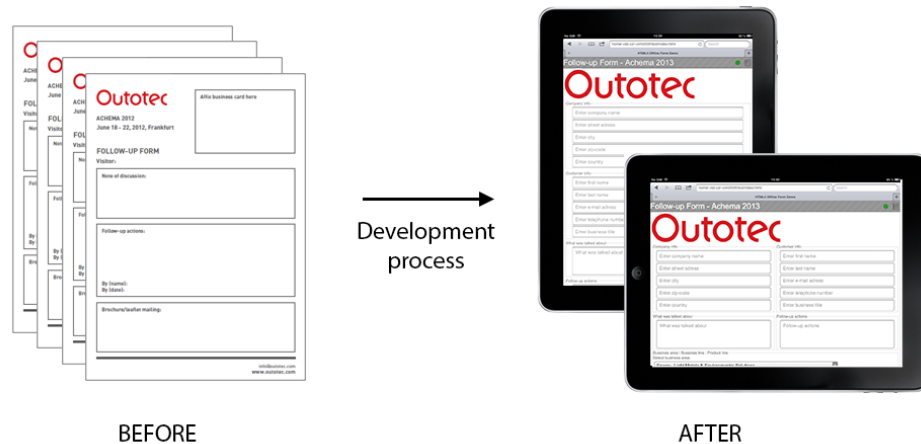


Figure 2 - Current and new Outotec Follow-up Form

### 2.1. Current system

Nowadays the Outotec Company is using an A4-size paper “Follow up” form for collecting the necessary data from customers during tradeshows (Figure 3). The form contains fields concerning customer’s company info, customer’s info, “what was talk about” field and description of follow up actions. The Current system requires printing out forms in advance which means costs for creating the form and also human resources costs for form processing after the tradeshow. Hand written filled forms are collected, sent to a worker who rewrites all data from paper forms to Outotec information system. Rewriting and processing the form described above can possibly cause damage such as entering corrupted data or losing a filled up form. There is also a relatively large time delay between filling the paper version and synchronizing these data with the corporate database. The current paper form processing is illustrated in Figure 4.

#### Advantages of paper form:

- Filling of the form does not require any power resources

- Does not depend on any electronic device or technology
- Paper form can be used as backup medium
- Virtual attacks resistant

**Disadvantages of paper form:**

- Time consuming form processing
- Resources consuming processing
- Processing comfort
- Possible loss of data / form
- Harm the environment – paper consumption
- Needs physical storage space
- Obsolete

Information Technology development led by portable devices such as smart phones and tablets enables us to use these devices in previously unthinkable ways. A tablet or a smart phone can be used as a tool for comfort data entering. Those data are safely saved in certain location and can be accessible by other information systems immediately after saving. The main advantage of using these devices is user comfort, no error rate during processing and the processing speed. On the other hand there is a possibility of device failure such as drain battery, internet connection or other technology problems. At present benefits of using electronic form filling and data collecting greatly outweigh disadvantages.

The image shows a 'Follow-up Form' from Outotec for the ACHEMA 2012 event. The form is titled 'Outotec' and 'ACHEMA 2012 June 18 - 22, 2012, Frankfurt'. It includes a box for 'Affix business card here'. Below this, it says 'FOLLOW-UP FORM' and 'Visitor:'. There are three main input areas: 'Note of discussion:', 'Follow-up actions:', and 'Brechure/leaflet mailing:'. At the bottom, there are fields for 'By (name):' and 'By (date):'. The footer contains the contact information 'info@outotec.com' and 'www.outotec.com'.

**Figure 3 - Current Outotec Follow-up Form**



### Important paper form's fields description:

- **Business card field**

It is expected from a customer to attach customer's business card to the form because of need of customer's information (name, address, company name, et cetera). It is also possible to write this information by hand.

- **Note of discussion field**

The field describing what was talked about during the tradeshow.

- **Follow-up actions**

The field contains description of follow-up actions including a responsible person.

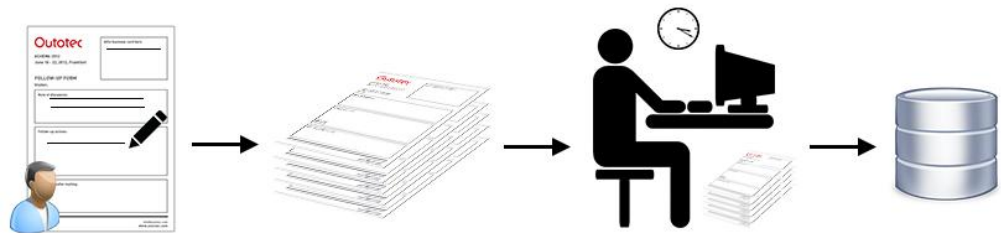


Figure 4 - Current system workflow

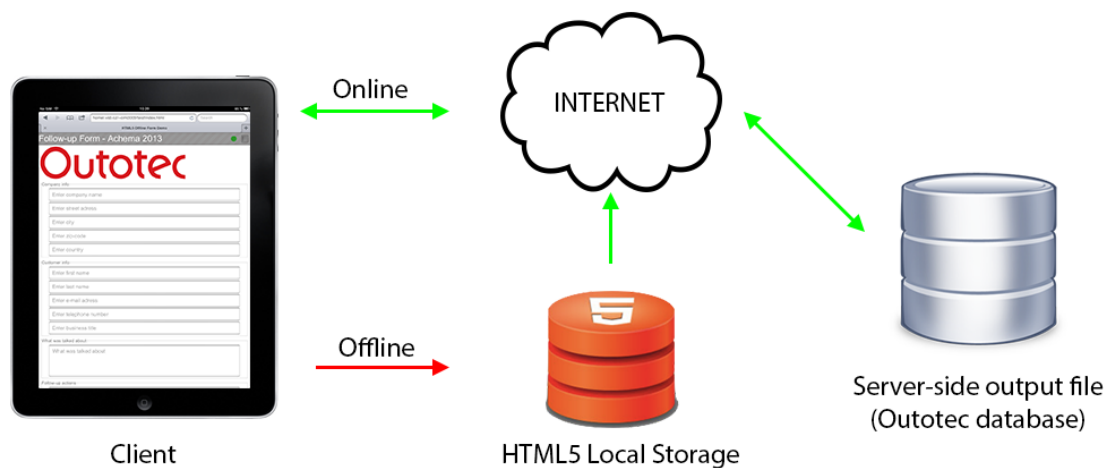
## 2.2. Required system

The main task is to develop a user-friendly electronic version of the Outotec's follow-up form which can be used with iPad or other tablet. Electronic form will contain all necessary fields to fully replace the current paper version with no data loss during the future filling. The main requirements for the development are simplicity, functionality and user friendliness.

### Technical requirements:

- Form filling
- Form submitting and saving
- Displaying and editing of previously saved records
- Possibility to use the system in offline mode
- Optimization for an iPad

The main idea of using this system is as follows. The customer fills in and submits all required data using Outotec's iPad. The System validates entered values and saves the data to the server. If internet/server connection is not available, the system will use a temporary local storage for data saving. Data from the local storage are sent to the server immediately when the system comes back online. Working online also enables to list all previous records and edit them if needed. The described system is illustrated in Figure 5.



**Figure 5 - Working schema of target system**

### **3. Development**

Development is the basic pillar of every software development process. It is a part of programmers / developers where customer requirements are transformed into the final product using development tools, technology and software development process.

#### **3.1. Development tools**

##### **Hardware:**

- **Web server**

Web server is a term for a configured computer that delivers content of web applications and pages to the user's browser. For testing purposes "HomeL" server provided by Vysoká škola báňská – Technická univerzita Ostrava was used. This server enables to host web applications, which are based on HTML, PHP and JavaScript technology.

- **iPad**

Tablet iPad is a touch device produced by Apple INC Company which runs Apple's iOS operation system. Large market share of this device between other tablets is the reason why many web pages and other applications are being optimized for this device. The current version of iPad enables browsing web pages, shooting pictures and videos, listening music and running applications available in Apple Store. Its large touch screen provides user a great user experience and comfort.

##### **Software:**

- **PSPad**

PSPad is a freeware text editor containing tools for developers such as code syntax highlighting or spell checking which enables easier and faster development.

- **File Transfer Protocol (FTP) client**

Web application development usually is not taking place on the web server itself. It is needed to transfer all source files to a web server for

the final deployment. File transfer protocol is a standard network protocol which enables to copy files between host devices. FTP is widely used over all TCP networks including the Internet.

- **Dropbox**

Dropbox is file hosting services enabling synchronization of local files using a dropbox client. All specified files are stored and synchronized in cloud service, which means these files are accessible from anywhere by login in to the dropbox service. It also enables to share synchronized files across more devices. This feature can transform dropbox from a file backup service to a powerful team development tool. Every team member has access to up-to-date files in a specified folder. Dropbox can avoid inconsistency and errors during software development in team.

- **Web browser**

Browser is a computer program for communication with servers and presenting retrieved information for users in the readable format. Communication with servers is possible via protocols such as HTTP, FTP. Browsers can be divided into two categories - graphical and text. Graphical web browser enables more complicated formatting of web page, including images and videos. Usually it is possible to extend functionality by installing plug-ins (processing Java Applets, Flash animation et cetera). The most known graphical browsers are Google Chrome, Mozilla Firefox, Opera and Internet Explorer. Representatives of text web browsers are Links and Lynx.

## **3.2. Technologies**

The following technologies were used during implementation in the final application: HTML, CSS, PHP and JavaScript. This chapter provides description of the above mentioned technologies.

### **3.2.1. HTML**

One of used technologies in the project was HyperText Markup Language, more known as its abbreviation HTML. The origins of this markup language are dated back to the 1990's, when British physicist Tim Berners-Lee was working

at CERN, the European Laboratory for Particle Physics in Geneva. He was looking for an easy way, how to share documents between him and his colleagues. This effort helped him to create a useful tool and laid foundations of the World Wide Web. During this effort, he had to accomplish several things (2).

It was necessary to create a new language, which enables formatting text and will be standard for all documents. It was not created from a scratch, but it was built on another well-known markup language SGML (Standard Generalized Markup Language). Documents created by this new markup language consist of many HTML elements. Every element in its general form has three parts

1. Start and end tag : `<title></title>`
2. Attributes within the Start tag : `<h1 attribute="value">`
3. Text content between tags : `<title> "Welcome Page" </title>`

Every tag has its own meaning, for example the "`<title>`" tag defines the title of the document, which is shown in the title bar. Tim Berners-Lee used a lot of similar tags from SGML, like `<title>`, `<p>` paragraphs or `<h1 to h6>` for different levels of headings. But one very important tag has been added by him. This tag has enabled linking between different HTML documents and has been named `<a>`. This tag has an important attribute "href", which takes the value of url address, where another HTML document is stored (`<a href="url">Text</a>`).

Presentation layer was another problem, which had to be solved. Tim Berners-Lee created a new browser, he called it World Wide Web, which can handle HTML documents and display them properly. It was not only the browser, but also an editor, which was very helpful in creating these kinds of documents.

Other problems concerning sending these documents through site have appeared. These problems have been solved by creation of transportation layer. This first primitive transportation layer consisted of server and HTTP (Hyper Text Transfer Protocol). The server was only handling HTTP requests and sending back responses, containing HTML documents, specified in the requests.

HTTP is a connectionless, stateless and media independent protocol for delivering all virtual data (files, images or anything else) on the World Wide Web. It is a request-response protocol between two devices. HTTP client is usually a web browser, which is sending requests to an HTTP server and accepting responses from this server. HTTP client also presents received data in easy-readable form for the user. As mentioned earlier, HTTP is:

1. Connectionless - connection is every time cancelled after each request/response
2. Stateless - client and server are aware of each other only during the requests, after that everything is forgotten
3. Media independent - it is possible to transfer anything with this protocol, only the special MIME (Multipurpose Internet Mail Extensions) type must be specified. Thanks to this information, HTTP client knows, how to handle the received data.

Block of HTML code - Source Code 1 shows the basic structure and syntax of HTML language. It is a simple webpage containing plain text "Hello World!".

```
<!DOCTYPE HTML> <!-- Information about used version of HTML -->

<html> <!-- this tag tells a browser, that this is HTML document
-->
  <head> <!-- head tag contains meta tags, information about used
scripts (JS, PHP), styles (CSS), title -->
    <title>Hello World!</title> <!-- title of the web page -->
  </head>
  <body> <!-- body tag contains visible part of HTML document -->
    Hello World!
  </body>
</html>
```

Source Code 1 - HTML code example

**Cookies** are a small amount of data, which WWW server sends to the browser, which will store them in the local computer. Cookies usually contain specific user related information like, when the user was connected last time or for example background color chosen by the user. Because the HTTP protocol is

connectionless, cookies are used for storing data, which can differ visit by visit of the web page (3).

**HTML5** is the state-of-the-art version of this language. It comes with many new features and optimization for smooth running on low-powered devices such as tablets or mobile phones. HTML5 is much more standardized than its ancestor. New tags have been created for dividing different content on the web page (<video>, <audio>, <canvas> et cetera).

#### **New Features:**

- Canvas - enables drawing 2D graphics
- Offline web applications - caching content of the web page for visiting page in offline mode
- Web storage - similar with cookies, also based on key-value pairs. Web storage has bigger capacity for storing large amount of data
- File API - new API for handling with files on user's computer

#### **3.2.2. CSS**

CSS was developed in 1997 for web programmers to split information written in markup language, from graphical interpretation. Because of the simplicity of web pages before the year 2000, CSS did not have a great success. Everything has changed, when web browsers began using more graphical elements than basic fonts and few colors for changing page backgrounds (4).

CSS has enabled keeping uniformity across all pages on concrete web site. It means that, all web pages can be linked to one CSS file, which will affect them all. CSS can set up many properties of each HTML element, like

- width and height of this element,
- absolute position of element to the screen or relative position to the parent element,
- colors - background color, font color, border color and so on
- margin, padding and lot of other properties.

With CSS it is also possible to set the look for different devices. This functionality is very important nowadays, when people can visit web pages from many devices with different screen resolution. This kind of web design is called responsive. Responsive web pages should provide viewer an optimal viewing experience from any kind of devices and easy navigation.

The newest version of Cascading Style Sheets is version three. CSS3 is still in the phase of standardizing, but web browsers are already supporting new functionality and features, which this version is coming with.

CSS3 contains many new features, which are worth to mention

- Border-radius - in CSS3 functionality for creation rounded rectangles was implemented
- Transformation - new possibility for making dynamic pages with CSS3 (rotating, moving, scaling)
- Opacity - making web elements transparent
- Media queries - enabling responsive web design. Web pages can look different on different devices

The project used one of the possible techniques to display content properly on tablets, when the user holds the table on portrait or landscape. For each of these possibilities a different layout is defined.

### **3.2.3. PHP**

At the beginning of this server-side scripting language this acronym meant Personal Home Page. It is a language designated for web development enabling dynamic creation of web content. Everything in PHP scripts happens on the web server, where the PHP module is installed. It means that a user who visits website can see only the results from these scripts. In HTML source code it is not possible to see anything from PHP scripts (5).

This language has been created by Rasmus Lerdorf in 1994. With PHP he could make his personal website more dynamic, for example processing data from HTML forms or connecting to databases for reading or writing the data. Later with growing popularity of this server-side scripting language, few other



programmers helped with implementing new functions for working with files, connecting to almost all known database systems or processing pictures. Also the meaning of this acronym has been changed, nowadays PHP is the recursive acronym "PHP: Hypertext Preprocessor" (6).

Advantages of PHP:

- Native support for many database systems
- Multiplatform
- Very good documentation for PHP functions

Disadvantages of PHP:

- Inconsistent in names of functions. For example functions for working with strings: strpos(), strchr(), str\_replace(), str\_pad(). The first two functions do not use the underscore, which can lead to confusion of programmer.
- Missing tool for debugging in the standard distribution. Programmer has to help himself in another way (using echo command et cetera)

Source Code 2 shows a simple PHP procedure for displaying "Hello World" text by a web server.

```
<!DOCTYPE html>
<html>
<body>
  <!-- PHP script included into HTML page -->
  <?php // This tag indicates start of PHP script
    echo "Hello World!"; // command echo prints string on standard
output
  ?>
</body>
</html>
```

Source Code 2 - PHP code example

JavaScript is a multiplatform interpreted computer programming language developed by Brendan Eich from Netscape Company. As the name can imply, that this language will be like Java, the opposite is true. From Java language it only takes naming conventions. It is more influenced by the language C (7).

Java language enabled programmers to make web pages more dynamic and interact with the user. It is not like the PHP scripting language, because JavaScript uses client-side scripts, so everything happens in a web browser. This ability enables quicker reaction for user's inputs. JavaScript functions can be included right into the HTML page or separately in another file with extension ".js", but the link to this file must be included into the HTML page. JavaScript can also interact with the Document Object Model (DOM), so there is an option to change anything on the web page. These changes can be triggered by user actions, for example user clicks somewhere on the page, JavaScript processes this event and writes x and y coordinates to the page.

Examples of usage:

1. Asynchronous updating web content without reloading the whole page
2. Animation of web page elements (resizing, moving et cetera)
3. Validation of user inputs

JavaScript functionality can be extended by many libraries. The most known library is JQuery, but it is possible to find smaller and less known libraries, which are specialized in only one technique (library for making photo galleries, pop-up windows et cetera)

Source Code 3 shows how to create a simple JavaScript function in HTML document, link this function to button event in a way that the function is called every time when the button is pressed.

```

<!DOCTYPE html>
<html>
  <head>
    <script language="javascript" type="text/javascript"> // This
tag indicates start of the script written in Javascript
      function helloWorld() // function, which is called after
clicking on the button
      {
        alert("Hello World!"); // alert throw a pop-up window with
specified string
      }
    </script>
  </head>
  <body>
    <button onclick="helloWorld()">Push the button</button> <!--
binding onClick event to our funtion. OnClick event is raised
every time, when user clicks on this button -->
  </body>
</html>

```

Source Code 3 - JavaScript code example

## AJAX (Asynchronous JavaScript and XML)

As the name can give a clue, AJAX (8) is not a single technology. It consists of many web technologies, which are brought together by JavaScript. The main asset of AJAX is communication with server asynchronously. This kind of communication has enabled programmers to change web content without reloading the web page (automatic updating of discussion, whisperers).

Incorporated technologies:

- HTML and CSS - presenting layer for data received by AJAX request
- Document Object Model (DOM) - enables change of any web element on the page
- Extensible Markup Language (XML) and JavaScript Object Notation (JSON) - format of transferred data
- XMLHttpRequest - for communication itself
- JavaScript - brought all these mentioned technologies together

## JQuery

JQuery (9) is a fast, small and feature-rich JavaScript library designed to simplify client-side scripting. It brings a lot of new features such as

- Improved DOM manipulation: newly can be used like CSS selectors, uses id and class of HTML elements
- Easier event handling
- Effects and Animation: provides function for making page visually interesting (fading up and down HTML elements, changing colors of html elements based on where the mouse pointer is, dropdown menus et cetera)
- AJAX

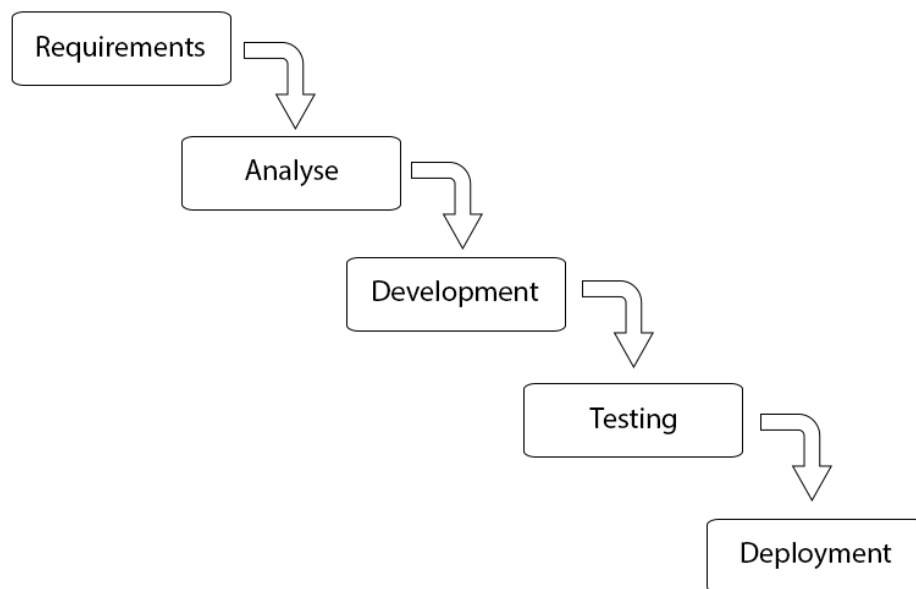
Source Code 4 shows how to use jQuery functions and link it to any HTML DOM object, such as item in the webpage.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Hello World</title>
    <!-- Link jQuery library to this page -->
    <script type="text/javascript" src="jquery.min.js"></script>
  </head>
  <body>
    <script type="text/javascript"> // Start of the javascript
    // function written in jQuery syntax
    $(document).ready(function(){
      // jQuery use DOM (find the html element based on its id)
      // with .html property can be set anything inside chosen tag
      $("#msgid").html("Hello World!");
    });
  </script>
  <!-- Div tag, which will hold text information -->
  <div id="msgid"></div>
</body>
</html>
```

Source Code 4 - JQuery code example

### 3.3. Development process

The new system was created using mainly waterfall development process, described in Figure 6 - Development process description. All requirements were specified by Outotec Company before the development process started. Requirements were analyzed and some of them lightly adjusted to fit the used technology. Development itself based on the previous detailed analysis was done in the team of two cooperating people supported by Outotec Company's employee. Small system changes were implemented as fast as requirements changed. The final application was tested by both developers and customers to debug the system before the final deployment. Deployment of application was done by Outotec Company according to its privacy policy. The used development process is described in Figure 6.



**Figure 6 - Development process description**

The process described above, also known as Waterfall Development Process is simple to use, but requires all target system requirements in advance. After every significant change of requirement it is necessary to start this process again from beginning.

## 4. Follow-up form application

This chapter describes the final application, the implemented features and the final deployment package including description of important files in this package.

### 4.1. Form description

Figure 7 shows the described screenshot of the final application in landscape layout.

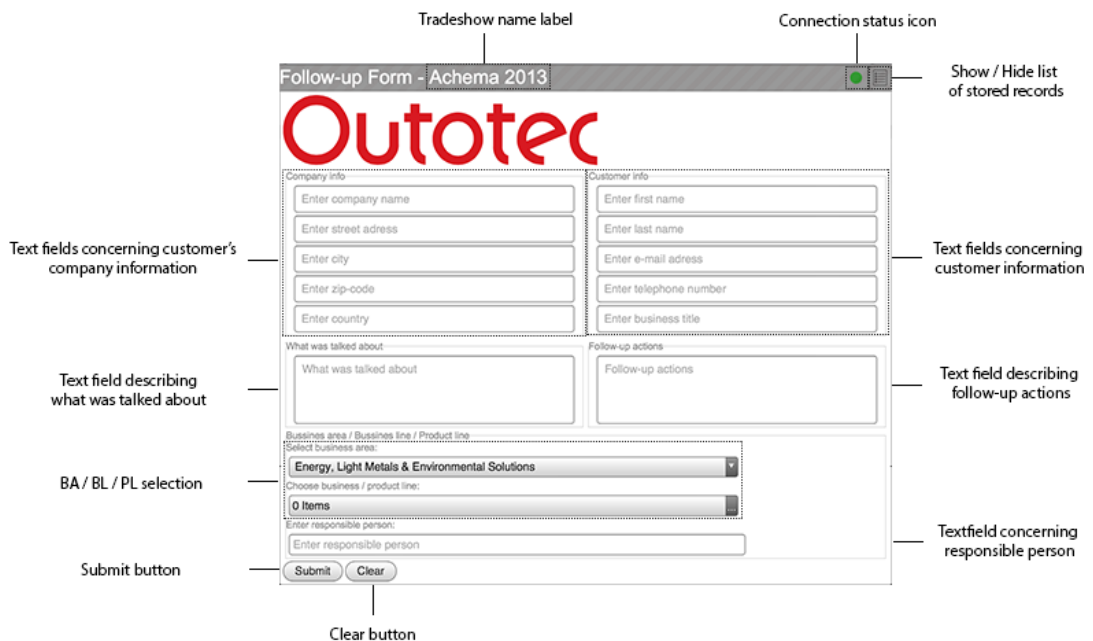


Figure 7 - Description of Follow-up Form

The final application consists of several important areas:

#### Tradeshow name label

It displays the name of the current tradeshow. The value of this label is stored in browser's cookies and it has to be entered due to the first form loading. Value can be changed by tapping (clicking) on current tradeshow name.

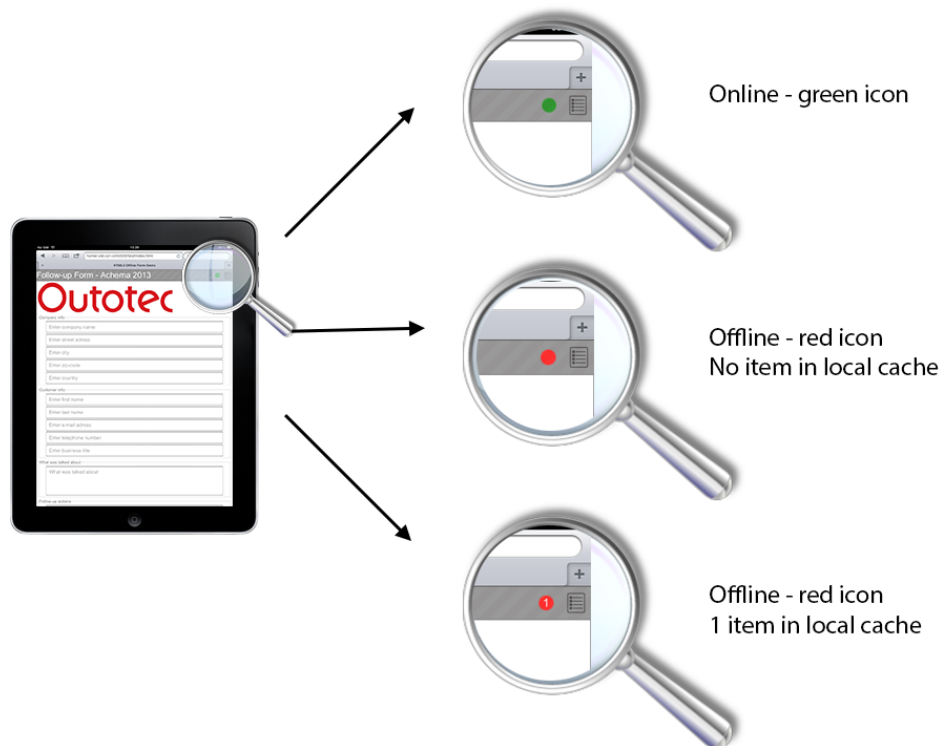
Used HTML tag: <label></label>

#### Connection status icon

It indicates the current network connection status. There are 2 possible states:

1. Online – indicates there is server connection available (green color)
2. Offline – indicates there is no server connection available (red color)

Offline mode also shows the amount of items stored in the application cache waiting for synchronization with server. All possible states are illustrated in Figure 8.



**Figure 8 - Form network connection statuses**

### **Show / Hide list of stored records**

Button is providing functionality of displaying and hiding list of the stored records. This is available only in online mode and it works as follows: the first click shows the list, the second click hides the list. The same can be done by swiping two fingers from side to side in touch screen capable devices.

### **Text fields concerning customer's company information**

Input fields concerning company name and address. Values from these fields are stored to output file after form submission. There are no mandatory fields in this area.

Used HTML tag: `<input></input>`

### **Text fields concerning customer information**

Input fields concerning customer's personal information such as name and contact information. Values from these fields are saved after form submission.

Mandatory fields: Telephone number or email address

Used HTML tag: `<input></input>`

### **Text field describing what was talked about**

Multiline input text field for description of customer's areas of interest. Short description of what was talked about during certain tradeshow. Value from this field is saved after form submission.

Used HTML tag: `<textarea></textarea>`

### **Text field describing follow-up actions**

Multiline input text field for description of follow-up actions. Information containing description of the next business steps for Outotec Company. Value from this field is saved after form submission.

Used HTML tag: `<textarea></textarea>`

### **BA / BL / PL selection**

Business Area field enables to choose one of the possible values. Depending on selected value options in Business Line / Product Line the field will change. Business Line / Product Line field enables to choose multiple options. All selected items (Product Lines) are saved after form submission including their Business Area index.

Used HTML tags: `<select></select>`

### **Text field concerning responsible person**

Input field containing name of responsible person from Outotec Company. Value from this field is saved after form submission.

Used HTML tag: `<input></input>`



### **Submit button**

Submit button is for form submission. After pressing the button mandatory field's values are checked and saved by saving procedure if passed.

Used HTML tag: `<input type="button"></input>`

### **Clear button**

Clear button enables to clear values from all fields. It is for resetting form to its default setting. Also all forms selected for editing (if any) are unselected and changes are discarded.

Used HTML tag: `<button></button>`

## **4.2. Features**

This chapter contains description of all significant features implemented in the final solution.

### **4.2.1. Data saving procedure and client-side local storage (app cache)**

#### **Motivation:**

In order to enable follow up form to work offline it is necessary to use resources, which provide possibility to store all necessary source files in the end device, when these are not accessible online. HTML5 brings the feature called client-side storage, which enables to store limited amount of certain files in browser's application cache, in the same place, where usually user specific information is stored, et cetera browser cache or preferences. Web application using this feature contains a manifest file including list of files, which are needed to be synchronized from server to browser application cache. Synchronized files are ready to be displayed even in offline mode and fully operational.

#### **Realization:**

The complete follow up form project consists of several files which are necessary for correct functionality. All these files are stored in webhosting

service and accessible by specific URL. To provide full functionality of the form all of these files are specified in cache.manifest file, which is stored in server as well.

**Use cases:**

1. Online

- a. The page is not stored in browser application cache.

Typical case, when user accesses the form/web page for the first time. Form will be displayed and the browser caches all files specified in cache manifest file to the internal cache.

- b. Page is stored in browser application cache.

If the cached files in the browser are consistent with files specified in cache manifest on server side, content is shown from cached files. If the cache manifest has changed since the last visit of the page, cached files are updated according to the new up-to-date version of the manifest.

2. Offline

- a. Page is not stored in browser application cache.

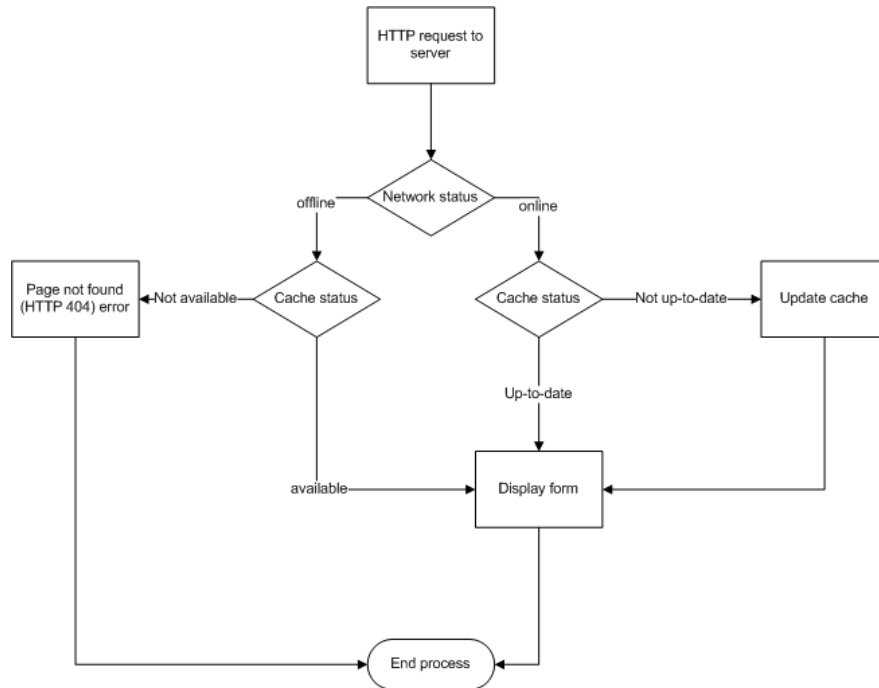
Page is not found and not displayed, because there are no accessible source files (no access to the server and no files in application cache). This case ends up with HTTP 404 error.

- b. Page is stored in browser application cache.

Server source files are not reachable, but all necessary files are stored in application cache from where are they displayed and processed.

**Workflow diagram:**

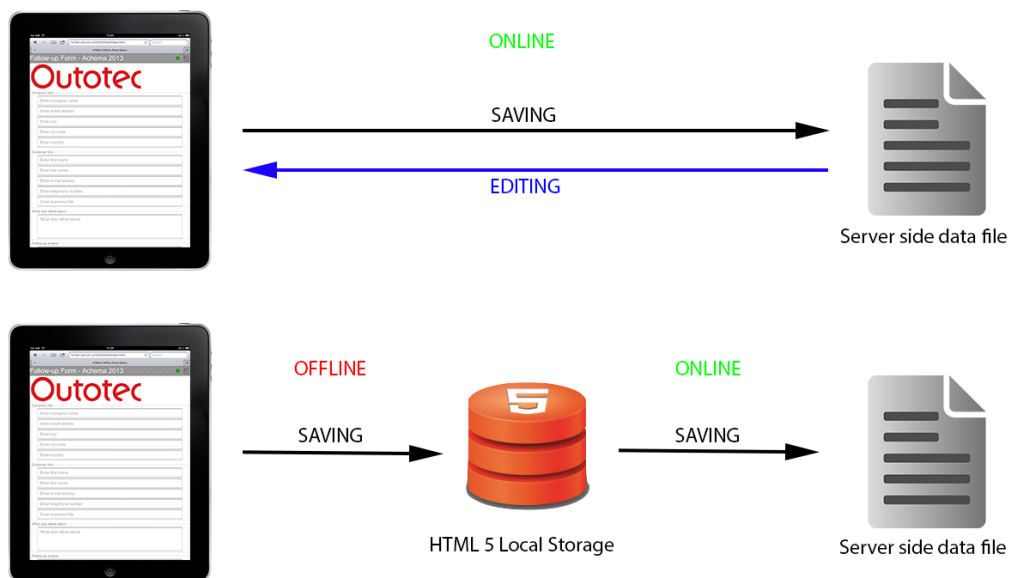
UML diagram in Figure 9 shows all possible options of saving form after user's submission.



**Figure 9 - Form saving procedure diagram**

**Example:**

Differences of form using in offline and online mode are illustrated in Figure 10.



**Figure 10 - Form saving procedure example**

#### 4.2.2. Form field auto complete feature

##### **Motivation:**

Outotec Follow-up Form contains fields and textboxes concerning customer's details such as names and addresses. Many of these customer's details are already stored in Outotec database, which enables to use these data for suggesting possible text field values based on previous user input. Selected option's data is filled in to the corresponding fields, such as company name and its address. This feature makes the form more user-friendly and saves time during filling in the form for current customers.

##### **Use Cases:**

1. User/customer is stored in database

User starts to write into text field, inside algorithm starts to looking for possible matches in intern database after every user change in the field. Possible matches are showed below the text box as in the list of possible values. After choosing one of them the current field is filled by the chosen value as well as depending fields.

2. User/customer is not stored in database

New user/customer starts to write into text field, but the algorithm is not able to find matching data in the database. In this case it is necessary to fill in all the data manually field by field, information about a customer is stored in the database and ready for the next time use with auto complete feature.

##### **Realization:**

Because of the requirement to be able to use the follow up form also offline without internet connection it was necessary to solve this feature by *JavaScript*. JavaScript enables to load stored file, parse this file, filter the results based on user input and also show or hide CSS element, which contains suggested values for the current field.

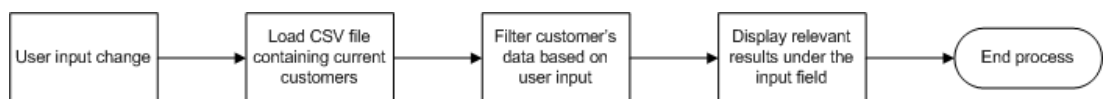
JavaScript function bonded to a certain text box field is called on every text box value change event. This function works as follows:

1. Processing source data file  
 This step includes opening and loading source data file containing customer information stored in CSV (coma separated values) format. These functions are part of the open JavaScript library jQuery. This library also enables to process and parse CSV files to arrays or different data structures, which are easier to filter.
2. Filter results based on the user's input  
 The array of possible options stores details only from customers where starting substring of certain value equals value entered by the user. The array of possible results is refreshed every time, when user changes input. Refreshing ensures the up-to-date list of possible suggestions.
3. Display list of matching results below the textbox  
 Non empty array of results is shown as CSS div element below the active input field/text. Empty list is not shown at all
4. After selecting one of the options the fields are filled by selected values.  
 Choosing one of the displayed options from suggestion box fills the active text box by selected value. It also fills some related fields as well.

In the case, when the customer is not stored in the cached file yet, it is necessary to enter all values manually. New values are automatically saved to the file with current customer details after form submission. Next time it will be possible to see this data in suggestion box.

**Workflow diagram:**

UML diagram in Figure 11 describes the workflow of whispering procedure initialized by user input.



**Figure 11 - Whisperer feature workflow diagram**

**Example:**

Figure 12 shows the whisperer feature implemented in a real device.



Figure 12 - Whisperer feature example

### 4.2.3. Editing stored records

#### Motivation:

Sometimes, there is a need to display previously stored records for many reasons such as editing or records viewing. HTML5 does not allow loading data directly from browser application cache, which is the reason, why this feature is available only in online mode. Using form online enables to display a list of stored records, select a specific record and edit it. All changes are saved to the output data file.

Touch capable devices such as tablets and other computers using touch screens bring revolution in application control. Using fingers and gestures to human is more natural than any other input device.

Swiping two fingers from side to side displays or hides the list of stored records. It is also possible to do this using a button from application menu bar in case of using a not touch capable display. Using more than one finger avoids unwanted displaying and hiding of list of record during form filling by accidental swiping.

#### Use Cases:

##### 1. Online

A user swipes two fingers from right to left side to display a list of records. Selecting one record loads data from a stored file to the form fields. The record is ready for editing and saving. The updated record is saved back to the output file. Swiping two fingers from left to right side of the screen hides the list of records.

##### 2. Offline

Displaying of list of records is disabled in offline mode.

### Realization:

HTML provides an access to device's touch events such as "touchstart", "touchmove" and "touchend". The application gets data from user's touches and gestures by firing the above mentioned events after every user input and it calls bonded function to do the following actions such as displaying or hiding list of the records. Selecting a record from the list of the records for editing calls JavaScript function, which loads all data from the selected record and fills this data to relevant form fields. All fields are editable and any changes saved by standard saving procedure back to the output file.

### Workflow:

UML diagram in Figure 13 illustrates the workflow of displaying list of stored records.

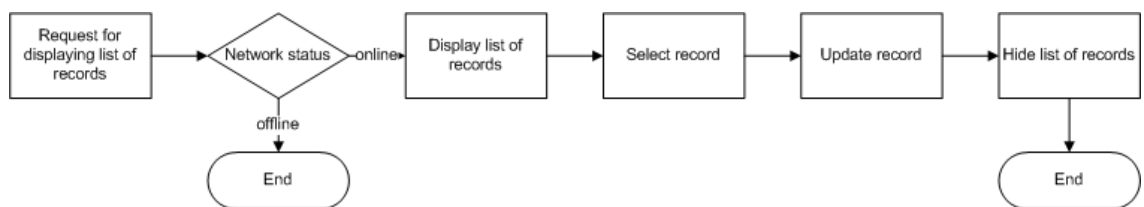


Figure 13 - Edit record feature diagram

### Example:

Figure 14 illustrates how to display, select and hide list of records in a real device.

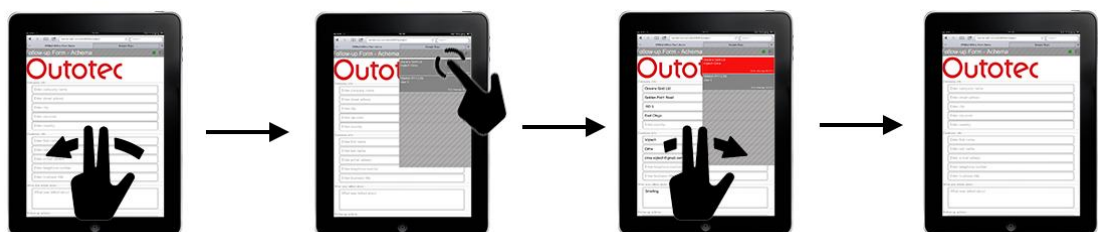


Figure 14 - Edit record feature example

#### 4.2.4. Dynamic form layout feature

Current tablets are using both landscape and portrait modes. For better user experience it is necessary to change the form layout depending on which

direction the device is used. The user can change the layout from portrait to landscape mode (or opposite way) by rotating the device 90 degrees.

The current version of CSS enables to use separate layouts and styles for each of these modes. This feature provides a possibility to adjust web pages and application for changeable screen sizes and ratio between screen width and height.

### Realization:

CSS file containing all form's item style and design is separated into 2 areas. The first one specifies styles for form in portrait mode, the second one specifies styles for landscape mode.

### Workflow:

UML diagram in Figure 15 describes the process for choosing suitable layout based on device orientation.

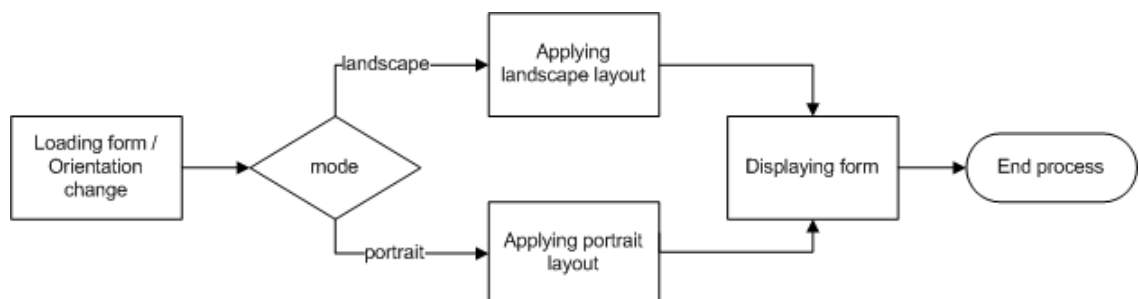


Figure 15 - Dynamic layout feature diagram

### Example:

Figure 16 shows how the form layout is changed based on device's orientation.



Figure 16 - Dynamic layout feature example



#### 4.2.5. Tradeshow name setup

Tradeshow name is one of the constant values during one event. To make the follow up form more user friendly, it is not required to fill this value every time, when the form is being filled by the customer. Tradeshow name value is set up only once and then stored in browser's cookie. It can also be changed by the user by tapping the current tradeshow's name.

Use Cases:

1. The user opens follow-up form for the first time (there is no cookie containing current tradeshow name). The form displays a popup window requiring tradeshow name. After submitting the tradeshow name is saved to cookie and shown in the menu bar.
2. The user taps (clicks) on the current tradeshow name to change its value. The form displays the popup window requiring tradeshow name. After submitting the new tradeshow name is saved to current cookie and shown in the menu bar.

**Workflow:**

UML diagrams in Figure 17 describe all possible ways how to fill in and store tradeshow name information.

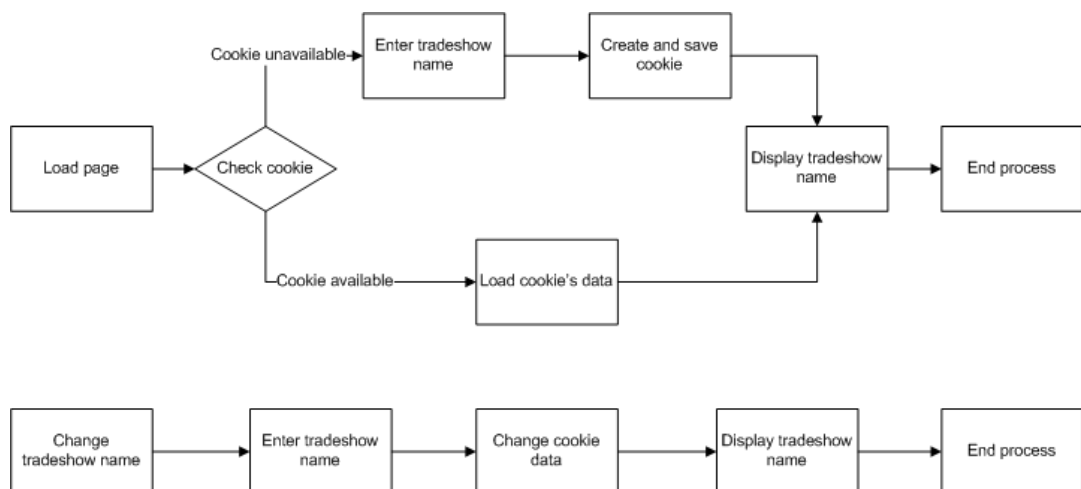


Figure 17 - Tradeshow name setup diagram

**Example:**

Figure 18 illustrates changing of tradeshow name in a real device.



Figure 18 - Tradeshow name setup example

### 4.3. Deployment package description

The final application deployment package consists of the following files:

- index.html – contains all HTML code
- styles.css – contains cascading style sheets for application design
- scripts.js – contains all JavaScript functionality (client-side functions)
- fileOperations.php - contains PHP functionality (server-side functions)
- Source data files:
  - companies.txt – contains source data for whisperer (field auto complete) feature
  - businessareas.txt – contains names of current Outotec’s business areas, product areas and product lines.
- Configuration files:
  - .htaccess – contains specification for browser data caching
  - cache.manifest – contains configuration for using HTML5 internal local storage
- Images
- data.txt – output file containing all stored data.

#### 4.3.1. businessareas.txt file specification

Source file businessareas.txt provides data to Follow-up Form elements concerning selecting business area, product areas and product lines. Records are stored in CSV format separated by new line (“\n”). Each record contains free fields separated by comma sign (“,”) in the following order: 1. Index of business area, 2. Product area flag, and 3. Product line name.

The index of business area is a number specifying item's business area. In the current system values are specified in Table 1.

Value	Business area
0	Energy, Light Metals & Environmental Solutions
1	Ferrous Solutions
2	Non-Ferrous Solutions

**Table 1 - Business area values**

Product area flag is a number value determining whether a particular record is the name of the product area or product line itself. Flag values are shown in Table 2.

Value	PA / PL
0	Product area
1	Product line

**Table 2 - Product area and Product line values**

Product line name is a string value containing the name of the product area or product line.

**Format of businessareas.txt record:**

index\_of\_business\_area, product\_area\_flag, product\_area-line\_name

**Example of businessareas.txt file:**

```
0,0,Energy
0,1,Oil Winning
0,1,Energy Products
0,0,Light Metals Fluidized Bed
0,1,Alumina Refinery
1,0,Ferrous Beneficiation
1,1,Beneficiation
1,1,Physical Separation
```

File example described above creates the following tree structure:

- Energy, Light Metals & Environmental Solutions
  - Energy
    - Oil Winning
    - Energy Products
  - Light Metals Fluidized Bed
    - Alumina Refinery
- Ferrous Solutions
  - Ferrous Beneficiation
    - Beneficiation
    - Physical Separation

#### 4.3.2. companies.txt file specification

File companies.txt contains data of companies which are already stored in the database. This data are used as a source for auto complete (whisperer) feature during form filling. Individual records are separated by new line (\n). Each line contains items separated by a tab sign (“\t”) in the following order: 1. Company Name, 2. CSN, 3. Address Line, 4. City, 5. Zip and 6. Country.

##### Format of companies.txt record:

Name	CSN	Address	City	Zip	Country
(Name/tCSN/tAddress/tCity/tZip/tCountry)					

#### 4.3.3. data.txt file specification

File data.txt is a server side output file that contains all saved data from filled forms. Records are separated by a new line (“\n”). Each record contains items separated by a tab sign (“\t”) in the following order:

1. timestamp – contains time of submission in milliseconds
2. tradeshow – contains name of the tradeshow
3. csn – contains unique company ID
4. company-name – contains company name
5. street-adress – contains company address
6. zip-code – contains company zip code
7. city – contains company city

- 8. country – contains company country
- 9. first-name – contains customer first name
- 10. last-name – contains customer last name
- 11. email – contains customer email
- 12. tel – contains customer telephone number
- 13. business-title – contains customer business title
- 14. discuss – contains description what was talk about
- 15. follow – contains description of follow-up actions
- 16. responsiblePerson – contains name of responsible person
- 17. areas – contains selected product lines

**Format of data.txt record:**

```
Timestamp tradeshow csn company-name street-adress zip-code
city country first-name last-name email tel business-
title discuss follow responsiblePerson
```

## **5. Conclusion**

During our work we have created the web application which has met all the requirements and can be used as a full replacement for the current Outotec Follow-up form. We have learned a lot of new features and state-of-the-art technologies, especially HTML5, CSS3, JavaScript and PHP. Many of these features, such as HTML5 local storage or touch events and adjustable form layout in CSS, were used during implementation. Newly acquired knowledge and technical skills are great for our future use in IT departments.

We have got many experiences from corporate environment, seen existing development processes, participated in a real company problem and we could be part of a team solving this problem. Our teamwork skills indispensably increased during implementation. This work also provided us many social and presentation skills during regular meetings and brainstorming.

We have got familiar with software development process from its basics through implementation and testing to final application and its deployment.

Working for Outotec was great experience with much useful information about technology processes, not only from IT area.

## References

1. *Outotec* [online]. [cit. 2013-04-17]. Accessed by: <http://outotec.com/>
2. *w3schools: HTML Tutorial* [online], [cit. 2013-04-17]. Accessed by: <http://www.w3schools.com/html/>
3. *HTML5 Rocks* [online], [cit. 2013-04-17]. Accessed by: <http://www.html5rocks.com/en/>
4. *w3schools.com: CSS Tutorial* [online], [cit. 2013-04-17]. Accessed by: <http://www.w3schools.com/css/>
5. *w3schools.com: PHP Tutorial* [online], [cit. 2013-04-17]. Accessed by: <http://www.w3schools.com/php/>
6. *PHP: Hypertext Preprocessor* [online]. 2001 [cit. 2013-04-17]. Accessed by: <http://php.net/>
7. *w3schools.com: JavaScript Tutorial* [online], [cit. 2013-04-17]. Accessed by: <http://www.w3schools.com/js/>
8. *w3schools.com: AJAX Tutorial* [online], [cit. 2013-04-17]. Accessed by: <http://www.w3schools.com/ajax/>
9. *JQuery* [online], [cit. 2013-04-17]. Accessed by: <http://jquery.com/>

## List of Figures

Figure 1 - Outotec Company building in Lappeenranta .....	2
Figure 2 - Current and new Outotec Follow-up Form .....	3
Figure 3 - Current Outotec Follow-up Form.....	4
Figure 4 - Current system workflow.....	5
Figure 5 - Working schema of target system .....	6
Figure 6 - Development process description .....	17
Figure 7 - Description of Follow-up Form .....	18
Figure 8 - Form network connection statuses.....	19
Figure 9 - Form saving procedure diagram .....	23
Figure 10 - Form saving procedure example.....	23
Figure 11 - Whisperer feature workflow diagram.....	25
Figure 12 - Whisperer feature example .....	26
Figure 13 - Edit record feature diagram.....	27
Figure 14 - Edit record feature example .....	27
Figure 15 - Dynamic layout feature diagram.....	28
Figure 16 - Dynamic layout feature example .....	28
Figure 17 - Tradeshow name setup diagram.....	29
Figure 18 - Tradeshow name setup example .....	30



## List of Source Codes

Source Code 1 - HTML code example .....	10
Source Code 2 - PHP code example.....	13
Source Code 3 - JavaScript code example .....	15
Source Code 4 - JQuery code example .....	16

## List of Tables

Table 1 - Business area values .....	31
Table 2 - Product area and Product line values.....	31